ATOM USEF GROUP

Newsletter One Sept 80

Welcome to the first issue of the Atom User Group Newsletter. I hope that in this and future editions, I can bring you all that's new in the world of Atoms.

As this is a first issue, much - though hopefully not too much - will no doubt be wrong about it and many of you will have ideas about it which I haven't already had. Please tell me about them; unless you do, I can't improve it!

Many thanks to those of you who have joined - I hope there will be many more of you. Apologies to those of you who joined around the beginning of August and got a decidedly slow response - I was on holiday.

In this edition I hope to introduce the Atom User Group in more detail than in the sheet you should already have recieved. Thos who haven't - sorry - let me know and I'll send you one. Hopefully also I can provide something for all of you.

Finally, an appeal for information for the next newsletter and the library. I can only publicise what you tell me, so please do! Anything - programs, news, hardware, problems both solved and unsolved is most welcome. In particular, if you have any special interest(s) and would like to be put in contact with others in the same field, please let me know.

-----*-----

Contents of this issue

Appeal

Mr Paul Hardy, of 32 Westlands, Comberton, Cambridge wants to know of anyone who is writing Educational programs. Anyone else who is doing anything in this field, he'd like to hear from you.

Contributing Material

As stated elsewhere, material is welcomed for inclusion in the software library or in the newsletter. It would be gratly appreciated however if the following guidlines could be observed.

Programs I would like on cassettes - I will of course return them, but please keep a backup copy in case of accident. As I have to type them in again to my machine I would appreciate it if you could send listings of only the shortest programs. Along with this, could you also send a brief description of how the program works and how to get it to do things. If the program is reasonably short or if you have a printer, I would also appreciate a listing with the cassette, as some cassette players do not work with others, or do so unreliably, and I do not yet know if this is going to be a problem.

Articles may be on any reasonable subject, but please can I have your manuscripts in good time for issues of the newsletter. Also I dont want to spend two days deciphering Chinese - if your handwriting isn't so good, please type it or write it in capitals. It will almost certainly have to be retyped, so don't worry too much about layout just so long as it is legible.

Please note that submission of an article does imply a willingness to have it printed, so please dont complain if this happens ! Software is accepted for inclusion on the undesstanding that it is free of copyright or that the User Group will not be held responsible for breach of copyright due either to including it in the library, distributing it to members or for the use members or others may choose to make of it.

It is also the case that the Atom User Group may not be held respansible for any use or misuse of any material that it may provide, whether from the library, the newsletter or any other source, or any loss or damage its use or misuse may cause.

TV interface

Aproblem has been reported by a user who has been trying to use a TV set without a vertical hold control. He has found that the set will not lock on to the Atom's UHF output.

This is because the CRT control i.c used in the Atom, the Motorola MC 6847, is designed for use with the American 525 line, 60 Hz field standard TV system. A British standard part is not availiable, so any attempt not to use the 6847 would have meant a higher price or no graphics, which wouldn't have been so good.

Most sets with a vertical hold control should be able to accomodate this with no problems. If your set cannot accept the signal, it is probably not suitable for use with the Atom, although I would be interes ted to hear of anyone who has overcome this.

Remember that an old 405 line v.h.f set will not work under any circumstances with the Atom unless it is extensively modified, and is certainly not worth the effort. Dual standard sets should however work on u.h.f.

Efficient Use of Memory

Whatever amount of memory you have on your Atom, remember the First Law of Computing : " Any program will be larger than the availiable memory ". This article is intended to help those who have been told the dreaded ERROR 30 or ERPOR 248.

The first step is to sacrifice readability for space, and to shorten the 1 lenpth of keywords, and to decrease thesize of vectors and arrays. Every character saved in a keyword saves 1 byte, every element in a word vector or array 4 bytes, every element in a byte vector 1 byte, and every element in a floating point array 5 bytes. The removal of non-significant spaces will also save memory at the rate of one byte for each space removed.

Removal of lines by the use of multi statement lines can also be used to save memory. Every line removed can save two bytes if formed into a multi statement line - three bytes are saved by deleting the line, but an extra byte is gained from the necessary semicolon.

For instance to dlear the array QQ;

```
10F.N=0TO99;QQN=0;N.  is four bytes shorter than 10F.N=0TO99
                                                 20QQN=0
                                                 30N.
```

Note that the use of smaller line numbers does not save any memory as line numbers are stored as pure binary, two bytes per line.

The above techniques can save a surprising amount of memory, but if a fairly large chunk is required they are inadequate. In this case, finding more is somewhat dependent on the machine.

If the lower text space is in use, and the extension ROM is not (it may be present, but not used) there is an additional 256 bytes availiable from #2800 instead of #2900. This is normally used by the floating point variables, but beware - reset will corrupt the program, as will use of floating point variables! Insurance against reset can be made by keeping note of the contents of !#2900, and by restoring it after reset.

If you have a 12K Atom, and are not using the higher resolution graphics modes, a larger amount of memory is availiable in the upper text space than in the lower. The lower offers a maximum of 4864 bytes (5120 if the floating point is not being used) while the upper offers 5632. However, you lose the high resolution graphics.

Partially expanded Atoms may be able to benefit from the above comments. However, memory chips can be moved about, and relocating chips may be able to provide more space depending on both the application and the system configuration. DONT however move ics every time you run a different program, neither the ICs or the sockets will withstand it too long ! Remember also that to use the floating point there must be at least 1K in the lower text space.

Minimum configuration systems will have to be fed more memory if you cannot save enough by shortening keywords etc. Bad news for 12K machines though- if you cannot save enough, you need the external bus and more memory cards, which cost !

Remember too though the possibility of using multiple text spaces - in the ideal situation of 12K RAM, no floating point and no high resolution graphics, a total of 10752 bytes is availiable.

A final word about assembly language programs, as these tend to be truly amazing guzzlers of space. These can be written as self - contained modules with a minimum of external references, and written, tested and loaded independently. The few external references can be supplied later, and the whole program will never be in the machine (in source form) at any one time.

This is in fact the essence of structured prugramming, which whole books can be - and are - written on, and which languages like Algol and Pascal take full advantage of. That, however, is another very long story.

Backus Naur Form

This article is intended to give some idea of what Backus Naur Form (B.N.F) is and how it defines the syntax of a language. BNF is used in Chapter 26 of "Atomic Theory and Practice" to define the syntax of Atom Basic and assembler, and examples are taken from there.

There are two main parts to any computer language; the "syntax" which defines what may legally be written, and the "semantics" which describe the meaning of what is written. B.N.F. defines the syntax, not the semantics, of a language.

For instance, 10 A=A/B is snntactically correct, but may be semantically incorrect if B=0 as division by zero is meaningless.

Backus Naur Form defines a language in terms of 'objects*' which are comprised of other objects, which may alse be defined or literal. An example is the definition of the object <digit> which is

<digit>::= 0|1|2|3|4|5|6|7|8|9

This reads "the object <digit> can be any of the characters 0,1,2,3,4,5,6,7,8,9" Defined objects such as <digit> are enclosed, by convention, in triangular brackets; objects not so enclosed are treated as themselves. The symbol | is read as "OR" ; one of the alternatives must be true. Groups of alternatives can be joined by use of curly brackets.

In addition, the character ^ can be used. This means "the previous object may be repeated any number of times, including zero".For instance, positive number is defined as

<positive number>::=<digit><digit>^
such that <positive number> is less than 2147483648

This shows a limitation of BNF in that a description of some limitation or feature may take an inordinate amount of space in BNF, or even be impossible to describe. Such features are normally described in plain English.

As a simple example of BNF, the syntax of the FOR statement will be derived.

<for statement>::=<half for><sd> |<half for>STEP<expression><sd>
<half for>   ::=FOR<variable>=<expression>TO<expression>

Taken together, this tells us the FOR statement can take two forms shown below.
FOR variable = expression TO expression STEP expression
FOR variable = expression TO expression
Further research gives fairly easily what <variable and <expression> may be assuming it wasn't already pretty obvious!

Lastly, don't get confused by BNF. It is a useful tool forfinding the correct syntax in a particular case, but is not essential, particularly in an interpreted language like BASIC where debugging and the 'try it and see" approach is so easy.

*Note the word "object" is used here where "syntactic entity" is used in the Atom manual. These terms are synonymous.

OPERATOR PRECEDENCE

The precedence of the arithmetic/logical operators does not seem to be given anywhere in the manual, except in a rather involved form in the syntax definition in Chapter 26. The following is intended to clarify the rules.

Expressions are evaluated left to right in general, unless this order is overruled by the rules of precedence, which are:

1) Expressions within brackets are evaluated first.

2) Next, functions are evaluated (eg ABS)

3)The operations *,/,%,?,!,& are evaluated

4) Lastly, +,-,:,\ are evaluated.

Within brackets, the above rules are followed again. Note that in a logical expression, <,>,= are evaluated AFTER all arithmetic or bitwise logical terms have been calculated. Remember also that in a logical assignment statement, the right hand side must be totally enclosed in brackets, ie

```
    Q=(A=B) is legal, but
    Q=A=B   is not.
```

As an example then

```
A=B+C*D*ABSE  is equivalent to A=B+(C*(ABS(E)))
Q=S*C<=J&K:L  is equivalent to Q=((S*C)<=((J&K):L))
```

Remember that compound functions (eg ABSCOS) are evaluated right to left(COS before ABS) and that the unary operators + and - are treated as functions.

A Useful Subroutine

A useful subroutine in the current 8K ROM is to be found at location #F7D1. This one is one that Acorn don't tell you about or give you an indirect vector to, so it may change its location or its nature in future issues of the ROM.

This subroutine prints the string at the end of the JSR instruction that calls it until a negative byte (that is, one with the most significant bit set)is encountered. Control returns to this byte, which should either be a NOP (#EA) or another instruction with the most significant bit set.

An example call is

```
:LIO JSR #F7D1
] $P="HELLO FRED";P=P+LENP;[
 NOP;RTS
```

Execution of this routine causes the message

```
HELLO FRED
```

to be printed on the screen. Note also the way in which data (in this case, a string) can be included in a machine code program.

A Fast Renumber

The program described here is an improved renumber facility for the Atom. It differs from the renumber given in "Atomic Theory and Practice" in that it is much faster, being written in machine code, and runs in screen memory without the need for elaborate precautions to prevent it from becoming corrupt.

The program can be loaded and run from screen memory by the *RUN "RENUMBER" command. The current text space will be renumbered only.

It should be noted that the renumber routine only renumbers line numbers; it does not redefine the destinations of GOTO and GOSUB statements. This is not a real problem provided labels are used.

Also, the destinations of gomputed GOTOs or GOSUBs will not bechanged, as this would be impossible (consider INPUT S; GOTO S for example). If this type of GOTO is used (and it can be very powerful) some alterations after renumbering will be necessary.

The renumbered program starts at line 10 and is renumbered by tens. This can be altered by altering the constant on line 150 to the desired value (the starting line number is always equal to the value of the increment).

No detailed description of operation is given, as it is the same as that given in the Atom manual.

To Load

Care needs to be taken when creating the tape, as the program can be carrupted by unfortunate use of the screen. The PRINT statements in the listing ensure that the cursor is kept off critcal areas.

To create a tape then:

1) Enter the program as given, taking care to omit no PRINT statements.

2) Type "RUN".

3) A pattern will form in the first few lines of the screen. Make sure that subsequent movements of the cursor do not cause the screen to be overwritten, or to scroll. Type "*SAVE "RENUMBER"8000 8050 8000", and save on cassette.

4) To run the program, type "*RUN"RENUMBER"". The same pattern as formed during assembly will form on the screen, followed by a brief snowstorm as the program runs. The program in the current text space will then be renumbered.

RENUMBER Program Listing

```
10 DIM LL10
20 M=#80;P.Ø12''''Ø21
30 FORT=0TO10;LLT=-1;N.T
40 FORX=1TO2
50 P=#8000
60[
70 :LL0 LDY @0
80      LDA 18 Current text space renumbered
90      STA M+1
100      STY M  M is text pointer
110      STY M+3
120      STY M+2  M+2 is new line number
130 :LL3 LDY @1
140      CLC
150      LDA @10   Immed. value is increment; alter for other values
160      ADC M+2
170      STA M+2
180      BCC LL1
190      INC M+3
200 :LL1 LDA (M),Y  Get next character
210      BMT LL10   End of text mark - exit
220      LDA M+3
230      STA (M),Y  Update line number
240      LDA M+2
250      INY
260      STA (M),Y
270 :LL2 INY
280      BNE LL6
290      INC M+1
300 :LL6 LDA (M),Y
310      CMP @#D    Search for end of line
320      BNE LL2    still in text
330      CLC
340      TYA
350      ADC M
360      STA M
370      BCC LL3
380      INC M+1
390      JMP LL3
400 :LL10 RTS
410];N.X;P.Ø6
420 END
```

LOOPS AND EXITING FROM THEM

A point not immediately obvious at first about loops of the FOR...NEXT and the DO...UNTIL variety is the two methods of leaving them : the normal and the abnormal variety.

A normal exit occurs when the corresponding satement to the one which opened the loop is executed (NEXT in a FOR loop, UNTIL in a DO loop) and control passes through it. No problems are encountered in this case.

An abnormal exit is encountered when control is transferred outside the loop by a GOTO statement, and control is not later transferred back into the loop without executing the DO or FOR statement opening the loop. A problem in this case is that after the loop has been exited, the FOR or DO is still 'active' as far as the interpreter is concerned.

If further FOR or DO statements are executed, the interpreter will add them to to the loops that are still active having been abnormally executed, and a depth - of - nesting error (18 or 111) may be generated.

This may also happen if the same loop is executed repeatedly - for instance, a program which waits for a command from the keyboard, searches a table for the command with a FOR loop and abnormally exits when the command is found. In this case the program will work as expected for a while and then inexplicably 'blow up'.

The cure is to execute an 'always true' terminator before exiting. In the case of the DO loop, this is the statement UNTIL 1, and the FOR loop FOR A=B TO C STEP C the statements A=C;N. A . As an example, see below.

```
100 Q=-1; DO Q=Q+1
110 IF Q=200 GOTO 200
120 UNTIL V?Q=A
```

```
100 FOR Q=OTO  199
110 IF V?Q=A GOTO 200
120 NEXT Q
```

These suffer from this fault but

```
100 Q=-1;DO Q=Q+1
110 IF Q=200 UNTIL 1;GOTO 200
120 UNTIL V?Q=A
```

```
100 FOR Q=0 TO 199
110 IF V?Q=A Q=199;NEXT Q;GOTO 200
120 NEXT Q
```

do not.

For the FOR loop, an alternative is that when the NEXT statement of a FOR loop t that encloses the FOR loop that has been abnormally exited is executed, the inner loop is automatically closed, provided that the outer NEXT specifies the loop vatable.

This gives another solution

```
90 FOR P=1 TO 1
100 FOR Q=0 to 199
110 IF V?Q=A GOTO 200
120 NEXT Q
200 NEXT P ; REM 200 NEXT will not do
```

ATOM USER GROUP

NEWSLETTER 2

Welcome to this, the second (and larger) issue of the Atom User group newsletter. Since the first issue was produced, the size of the user group has gone up about five times, and at the time of writing stands at about 400 members. As you can see, I now have a printer (a Centronics 737) and I can now offer printouts of all programs in the software library. Details elsewhere in this issue.

Many thanks to all of you who have contributed information and programs - we now have a fair number of games programs availiable. Please keep them coming - the software library is one of the main services the user group sets out to provide, and it entirely depends on you to function at all.

Many of you have been complaining about poor responses from Acorn. I have been in contact with them and they say that this is because of the tremendous demand there has been for the Atom, and they have been overwhelmed. They say, however, that they are beginning to manage and should be able to give a reasonable response soon.

Lastly, I have moved to Coventry, and thus I cannot answer telephone calls at Newton Ferrers. I do not yet have a permanent address in Coventry, so please address all mail to Newton Ferrers, from where it will be forwarded to Coventry. I will circulate a permanent address as soon as I can.

Contents of this issue

Atom Power Supply

Users of older Atom will have come across a problem when trying to expand,the Atom power supply is not powerful enough.

This article describes a supply which is powerful enough for the largest Atom, and leaves a fair reserve for expansion.

Newer Atoms do not have this problem as the ACOM supplied PSU has been upgraded and low power RAM chips are now supplied. However this article may still be of use if you are trying to expand.

The circuit diagram is shown below. It is a simple full-wave rectifier and IC regulator. The transformer reduces the mains voltage to 18 Volts with a centre tap: D_1 and D_2 then rectify this to provide approximately 12V. C_1 smooths the waveform which is then regulated by IC1.

Construction is straight forward and layout is not critical. The main points to watch are:

1. Use heavy gauge wire throughout as the currents involved are quite high (up to 5 Amps).
2. Ensure the regulator is well heatsinked as it runs hot and inadequate heatsinking will cause the regulator to cut out. About 12 square inches of 18 gauge aluminium mounted vertically should be enough, but be generous.
3. Ensure C_2 is mounted close to ICI.

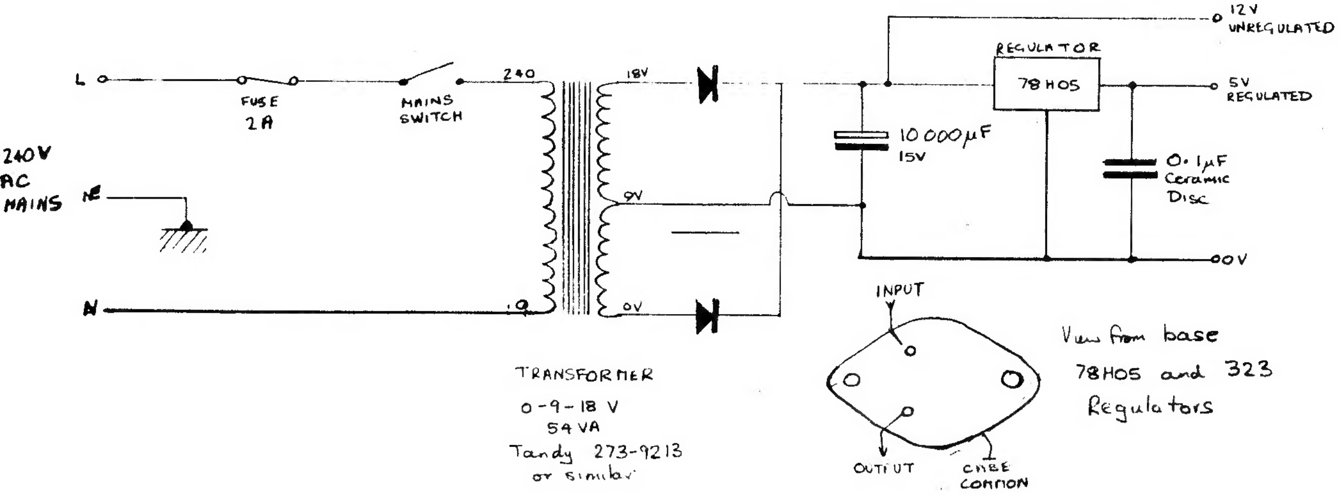
Components

No parts are critical and any similar parts may be used. C_1 may be anything over 4700 uF at 15 volts. The regulator may be 78HO5, 78PO5 or LM323 (note the last will reduce the maximum current to 3A). The transformer is a Tandy 273-9213. The diodes may be any rating over 50V, 5A.

Warning

As mains voltages are on parts of this unit, ensure that it is not possible to touch any live part of the circuit. If you use a metal box, ensure that it is well earthed.

ATOM POWER SUPPLY



Text Storage in the Atom

It can happen, when a program "crashes" that odd line numbers and other "junk" can turn up in a program and be imposs. to delete. This most often happens in Atoms where the upper text space only is in operation and you have been entering data onto the screen using "!" or "?" operators, but it can happen with machine code programs that have crashed or anywhere use of ! and ? operators is taking place. To recover, it is useful to know how a program is stored so some judicious "poking" can restore your program.

This sort of knowledge is also useful if you are writing programs which alter or access the program memory: good examples of this are RENUMBER and PACK.

Lines are stored in memory nearly as they are typed in(some BASICS "tokenise" keywords etc). However, Atom Basic does not. It does however store line numbers in a compressed form.

Any BASIC line when stored internally contains 3 items:

1. The Line number is the first item stored. This is stored in pure binary, most significant byte first (this is the opposite to the way the Atom normally stores numbers). Thus the line number 1020 is stored as # FC, #Ø3
2. The program text is stored exactly as it is typed in.
3. The line is terminated by a newline (#D). The next line number follows immediately.

The first byte of any program is #D (a new line). The line number of the first line of the program follows.

The last byte (TOP ? - 1) is a negative number. It immediately follows a newline. Thus the end of program is represented by a newline then a negative byte. This is why BASIC line numbers must be less then 32768.

A crash recovery program can use these rules to find where in memory has been affected and can then ? new data in to patch theerror (or make it accessible).

Expanding the ATOM (without ACOM's help ------)

With the one exception of the floating point ROM it is possible to do all the on card expansion for the Atom without bying ACOM's expensive memory chips. Here's how:

The RAMS used on the Atom are industry standard 2114 RAM's which are widely available and cheap (around £2 to £3 per 1C). The speed section used on the Atom is the slowest (450 nS) so any chips of 450nS or faster will do.

Older Atoms used the "normal" version but due to some problems (overheating of regulator and some 'sticky bits' especially in the upper text space) the newer ones use low power 2114s (2114L). The recommended type for expansion is thus 2114L, 450nS version. For instance, Technomatic are selling suitable IC chips at £2.25 each at present.

Expansion

It's largely up to you where you put your extra RAM chips, but it is wise to fill up in an organised manner. A suggested plan is (assuming you are starting from a minimal system).

✓ 1. IC40 and IC41 (Check that 1C30, a 74LS138, is fitted first. If not, fit it and cut the link in front of it).

This will treble your program memory available.

✓ 2. IC38 and IC39 -more programs memory still (2½K)

✓ 3. Ensure IC6 (74LS138) is fitted: if not, fit it. Then insert 1C10 and 11. This will give you 3/4K for program in the lower text space, and graphics up to mode 3. Longer programs can still be run by ? 18~~=~~# 82 NEW (converting back to the upper text space).

✓ 4. IC 12 and 13. This gives 1¼ K of lower text space memory

Then expand as your interest takes you - if you want high resolution graphics, fit IC 32 to IC39, otherwise fit IC 13-19.

Expanding the VIA and bus is even easier - just buy the bits and plug them in. Don't fit 1C50 unless you actually have a printer or something else you want to run off the printer port - it's not worth it.

The parts used are:

VIA	ICI	6522
	IC50	74LS244
Bus	1C2,1C3	81LS95
	IC4	DP8304 or INS 8208 (equivalent)
	IC5	74LS30

Connectors Bus 64 way Eurocard plug,
Printer 26 way Scotchflex header
No 3439 - 1302 (Matching socket 3399-6000)

Hardware Library

It is inevitable that many users will be 'doing their own thing' and designing their own circuits for fitting onto the ATOM. Many of these designes will be of interest to other ATOM owners and it has therefore been decided to setup a library of these deisgns. It will be run on the same general lines as the software library. Copies of the designs will be distributed free to members on request.

However it is necessary to set down some standards for documentation as it is not practicable for me to tidy up the documentation on designs, or even to check that they really work.

A provisional set of guidelines is as follows:

Drawings: These should be neatly set out on either A4 paper (for photocopying) or tracing paper in A0, A1, A2, A3 or A4 sizes (for printing). It is not possible to return tracings as they will be needed for reproduction in the future. However it will be possible to return prints. All components should be clearly annotated and all lines which go off-board should also be annotated.

Descriptions A brief explanation of any circuitry which function is not obvious should be included. A general description of the function of the circuit and of software required to drive it will also be needed. Especially remember to include special memory addresses etc used by the device.

Software A listing of any software needed to drive the interface should also be included. When producing the listing, please be generous with comments.

Testing It will not be generally possible for me to test designs as they come in, so please be sure that your drawings are up to date and you have tested your design and it works. Include a list of any special hardware you need to make it work.

I hope the above fairly rigorous standards to not put you off: however, they are necessary if the hardware library is to produce any useful assistance.

At present designs available include:

Senal Printer interface (described in this issue)

13K Static RAM Card (requires Atom with bars extension. Uses the rather expensive TMS 4044 4K x 1 static RAM so really of use only if you have a cheap supply of these. Extends lower text space to # 6 FFF. Fits on a single Eurocard).

KEYBOARD INPUT ROUTINES

Tim Jackson

During interactive programmes, the obvious way of inputting information is to use the INPUT statement. But this cannot be used during animated graphics sequences because it hangs around waiting for an input and the processor cannot get on with other things if no key is pressed. A less important disadvantage is that non-technical users have to be continually reminded to press RETURN after typing in a reply.

The latter problem can be solved by using the operating system OSRDCH routine. Perhaps the simplest use is:-

```
:
200 PRINT "PRESS SPACE BAR TO CONTINUE"
210 LINK # FFE3;  REM LINK OSRDCH
:
```

OSRDCH will wait for any key (except REPT, CTRL, SHIFT, LOCK, ESC, BREAK or a cursor control key) to be pressed before going on.

If you wish to detect any particular key, a machine code routine can beused to call OSRDCH, as follows

```
10 P = # 81
20   JSR  # FFE3 ;  STA #80 ; RTS;
```

This assembles machine code in locations #81 to #86. When called by LINK #81, it jumps to OSRDCH, and when a key is pressed the corresponding ASCII code is stored in location #80. An example call, for use at the end of a games programme:

```
1000 PRINT "WOULD YOU LIKE ANOTHER GAME?"
1010 PRINT "TYPE Y OR N"
1020 LINK #81
1030 IF ? #80 = CH "N" THEN END
1040 IF ? #80 = CH "Y" THEN RUN
1050 GO TO 1010
```

Once the machine code is assembled, it will survive the NEW command, and even BREAK, so it can be called from subsequently loaded programmes if desired. Since it is so short, an easy way to load the machine code is by executing.

```
! #81 = # 85FFE320;  ! #85 = # 6080
```

To input a fixed number of characters, say four, no doubt it would be possible to call the machine code from a FOR....NEXT loop of the required length. The programme would exit from the loop and continue as normal as soon as the fourth key was pressed. Alternatively, it should be possible to expand the machine code itself to include the required loop.

None of this allows keys to be used as controls during animated graphics sequences. The easiest keys to use for this purpose are SHIFT, REPT and CTRL, because as can be seen from the ATOM's circuit diagram they are separate from the matrix in which the rest of the keys are connected. SHIFT is connected to bit 7 at location # B001 (see p.194 of the manual) and thus ? # B001 & # 80 returns 0 if either SHIFT key is pressed, and # 80 otherwise. ? # B001 & # 40 and ? # B002 & # 40 act similarly for the CTRL and REPT keys respectively, except they return # 40 when no key is pressed.

As an example, try the following programme:

```
10      DO
20      X = ? #B001 & # 8Ø
30      Y = ? #B001 & # 4Ø
40      Z = ? #B002 & # 4Ø
50      If X = Ø   PRINT "S"
60      If Y = Ø   PRINT "C"
70      If Z = Ø   PRINT "R"
80      If X = #80 AND Y = #40 AND Z = #40    PRINT "N"
90      UNTIL Ø
```

A more useful (?) application might be to move the Starship Enterprise right when REPT is pressed and left when SHIFT is pressed, and to fire the phasers at the advancing Klingons when CTRL is pressed!

Detecting any of the other keys means looking at the keyboard matrix in the circuit diagram to see which keyboard row and column it is wired into. Then write the number of the row into location #BØØØ, and check the required bit of #BØØ1 to see if the key has been pressed. The following example checks for the Space bar:

```
10      ? #B000 = 9
20      DO
30      IF ? #B001 & 1 = 1 PRINT "NO SPACE"

40      PRING "SPACE"
50      UNTIL 0
```

? //B001 & 1 looks at column 0 of the keyboard matrix. If the space bar is press, ? //B001 & 1 will become 0, so line 30 will be skipped and line 40 executed.

The following example sets up a FOR....NEXT loop to scan the keyboard rows, and then looks at columns 1 and 2 of the keyboard matrix (lines 40 and 50 respectively):

```
 5      A = #B000 ; B = #B001
10      DO
20      FOR J = 0  TO 9
30      ? A = J
40      IF ? B & 2 = 0  PRINT 3-J
50      IF ? B & 4 = 0  PRINT 13-J
60      NEXT J
70      UNTIL Ø
```

With this program running, try pressing any combination of keys 0 to 9. The expressions in the PRINT statements in lines 40 and 50 calculate the relevant number from the number of the keyboard row (J).

Note that keyboard columns 3 and 4 can be looked at by ? B001 & 8 and ? B001 &16, but looking at column 5 is dangerous, as will be shown below.

Anyone who runs the first program above (to detect the space bar) will find that he cannot get out of the DO .oop by pressing ESCAPE. This is because the ESC key is in row 0 of the keyboard matrix, and is disabled by line 10. However, if you press key R the ATOM will be fooled into thinking you pressed ESC, because R is in column 5, like ESC, and is in the same row as the space bar.

This method cannot be used to detect any of the keys R to Z in column 5, because if you try it your programme will end prematurely!

Another drawback is that the graphics modes use higher bits of location #B000. So the CLEAR statement will set bits 0 to 3 of #B000 to zero, re-selecting row 0 of the keyboard; the row you want will have to be reset afterwards. But when selecting a keyboard row while using any of the high resolution graphics modes, care is needed or you will find yourself in a different graphics mode. For example, line 10 of the space bare detecting programme should be changed to:

```
          10     ? # B000 = ? # B000    9
```

performing a bitwise OR operation to preserve bits 4 to 7 of #B000. Note that if bits 0 to 3 started as anything but zero, a more complicated routine will be required - for example in the FOR ... NEXT loop in the number-detecting programme above.

A final point to watch is that you should reset bits 0 to 3 of location # B000 to zero before attempting to use any routine which itself performs a keyboard scan, such as the INPUT statement. Otherwise general consternation and changed graphics modes are likely to result!

* Or you can store it in one of the BASIC variables - see elsewhere in this issue.

NEW PRODUCTS

ACORN have recently announced the following program for the Atom:

Games Pack 1:	ASTEROIDS, SUB HUNT, BREAKOUT
Games Pack 2:	DOG FIGHT, MASTERMIND, ZOMBIE
Games Pack 3:	RAT TRAP, LUNAR LANDER, BATTLESHIPS
Games Pack 4:	STAR TREK, BLACK BOX, SPACE ATTACK,
Soft VDU Pack:	This is a set of programs for mixing text and high-resolution graphics.

Also announced are:

Atom PASCAL — a Pascal compiler for Atom in ROM. This plugs into the 8K ROM socket and allows you to swap between Basic and Pascal via a switch.

80 x 25 VDU card

32K dynamic RAM card

6502 ICE (in circuit emulator)

I don't have any information about availability of the above - contact Acom for details. The last 3 cards will require the external bus - they are really intended for the layer systems 2, 3 and 4 but should work on Atom, although there may be a lack of software for the 80 x 25 VDU and ICE.

A DOS and disk for Atom should be available in the first quarter of 1981.

A £75 Printer

Poundgate Electronic Design Services have informed me of a device they are offering for the Atom. It consists of a box which fits onto the Atom and drives a printer of the Creed 7B or 75 variety, which are widely available as surplus in the £30 up price bracket. These are plain paper printers which give a good quality upper case printout, although somewhat slowly and nosily. Poundgate supply the interface box and software for £35, or with a Creed 7B teleprinter for £75, or with the faster Creed 75 for £90. Details from Poundgate at Poundgate Farm, Beguildy, Knighton, Powys: Tel: 05477 273.

BACK ISSUES

As there appears to be a great interest in back issues of the Newsletter, I propose to introduce a back issue service.

All issues will be available. Back issues will be exactly like their originals, with the exception that out of date ifornation (old software library indexes etc) will be removed. The price of back issues will be 50 pence per copy.

Only Issue 1 has been circulated to date, so only issues 1 and 2 are currently available. Please state which you would like.

Useful Addresses in the Atom

Since the last issue, a number of useful addresses in the Atom have been unearthed by inquisitive users. The ones I know about are listed below.

Please note that as these are not ones ACOM publish,they may change in the future, so software using these addresses will not necessarily function if a new ROM is ever issued. If you contribute programs using any of these, please state that you are using such-and-such an address .. it may help me in the future if a new ROM is issued.

Addresses of stored variables

The Basic variables are @ - Z are stored in location # 321 to # 38C inclusive. Any variable is not stored contiguously so it cannot be accessed by the ! operator. Fig 1 gives the address of each byte each varable (MSB - most significant LSB - least significant).

Fig. 1. Addresses of variables

Variable	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
LEAST SIG BYTE	321	322	323	324	325	326	327	328	329	32A	32B	32C	32D	32E	32F	330	331	332	333	334	335	336	337	338	339	33A	33B
NEXT BYTE	33C	33D	33E	33F	340	341	342	343	344	345	346	347	348	349	34A	34B	34C	34D	34E	34F	350	351	352	353	354	355	356
NEXT BYTE	357	358	359	35A	35B	35C	35D	35E	35F	360	361	362	363	364	365	366	367	368	369	36A	36B	36C	36D	36E	36F	370	371
MOST SIG BYTE	372	373	374	375	376	377	378	379	37A	37B	37C	37D	37E	37F	380	381	382	383	384	385	386	387	388	389	38A	38B	38C

All addresses in hex

Relocating Arrays Arrays in the Atom are usually stored immediately after the end of the program text (offer TOP). If you have a long program with a lot of array space allocated, and you are not using graphics you can locate the arrays in video memory (or anywhere else for that matter) by changing the data in locations 36 and 35 decimal.

For instance, to locate arrays at # 8200 in the video memory:

?35=0; ?36 = # 82 before dimensioning arrays. This will give more text area in the lower text space.

For the technical location 35 and 36 appear to be the "free space pointer" described on p.145 of Atomic Theory & Practice.

TOP is stored at location 13 and 14 decimal if it's any help ------

COUNT appears to be stored in location 7 in case you want to get at it from assembly language prgram.

The BASIC Line Editor is at location CDØF. From a machine code program, a JSR # CDØF will

a). Print the character in A as a prompt

b). Wait for the user to type in a line of text. This will be stored from location # 100 onwards. A maximum of 63 characters may be input. A carriage return terminates the line and returns control to the caller. An ESC returns control to Basic. The normal line editor commands - RUBOUT and CTRL/X - do their normal action of deleting the last character and the whole line.

A routine for printing a constant string is located at F7D1. This can be called from an assembly language routine by JSR #F7D1, and it prints the characters following the JSR until a negative byte is found. It then returns and executes the negative byte, which would normally be # EA (NOP)

```
Eg. JSR  #F7D1
        $P = " HELLO ALL", P = P + LENP;
              NOP
```

This would produce

```
HELLO ALL
```

to be typed on the VDU.

Hex Output is provided by a routine at #F802. This routine types the number held in A in hex. A is the only register destroyed (along with status).

Machine Code programs can be loaded of tape and run instantly, with the *RUN command, but ACOM don't tell you how to do it with a BASIC program. It is however possible.

To do this, save the program as follows

```
   P. & TOP
say    3BC7 > * SAVE  "PROG"  2900  2BC7  CE86
```

This saves the program normally, but replaces the "normal" BASIC entry point of C2B2 with the new entry point CE86.

To run this program, type
* RUN "PROG"
and the program will be automatically loaded and run.

A problem occurs however if a DIM statement is executed - an error 30 results, if anyone solves this one please let me know.

An alternative method of running a program in a different text space is ! 5 = < address of text space 7 LINK # C2F2. This has the advantage that the program does not need to be on a 256 byte boundary.

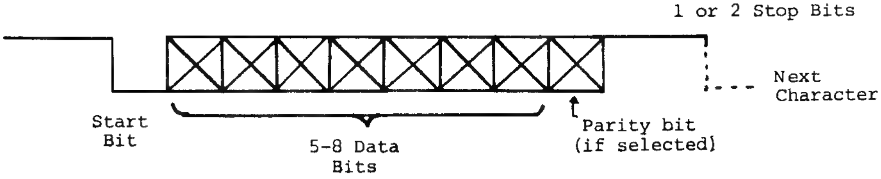
The cursor may be manipulated by altering addresses DE and DF, which contain the address of the cursor. The character used for the cursor is stored in location # E7.

A Senal Printer Interface

The ınterface described here is intended to enable users of the Atom who have a serial printer of the Teletype or similar variety to use it with the Atom's printer port, which is of the Centronics type.

It has been designed so that use of serial printer is transparent to the Atom, that is the serial printer looks, to the Atom like a parallel printer. This is so that the Atom printer controls - STX and ETX characters (2 and 3) - control the printer without any extra software being required.

The circuit is shown overleaf, and operates as follows. The UART chip, 1M6402, is an LSI device intended for use in serial asynchronous communications systems. It produces a serial output in the form shown below:



The start bit tells the receiver that a character is coming. It is always a 'low' and the start of a new character is given by a 'high' (the inactive state of the line) to 'low' transition at the transmitter output. This is followed by up to eight bits of data, which are sent least significant but first, and one, 1½ or two stop bits which signal the end of a transmission In addition, to improve the sensitivity of the communications link to errors, a parity bit may be inserted after the data bits and before the stop bits. There are two sorts of parity: even and odd parity. In even parity, the total number of '1' bits in the data and parity bits is always even: ie if the number of '1's in the data is even, the parity bit is Ø: if it is odd, the parity bit is a 1. With odd parity it is the other way round.

The UART consists of two sections: a transmitter and a receiver. In this design, the receiver section is not used but is described for completeness .

On receive the UART waits until it receives a "1" to "O " transition on its receive input (RRI) pin. It then waits half a bit time and then samples the pin again. If the input is still low, it starts to read a character. Otherwise it resumes waiting for a "1" to "O" transition. This helps it to ignore noise on the line.

After detecting the start bit, it waits for one bit time before sampling the input signal again. It reads the state into the receive register and waits for the next bit to come in.

When all data bits have been received it checks the parity if this has been selected. If there has been an error, the parity error output, pin 13, goes high to indicate this. The UART then looks for stop bits ("high" input) and if it does not find them, the "framing error" flag goes high. The "receive"register is then transferred into the receive buffer register, where it is available to the external world.

On transmit the UART transmits data in the same format, automatically appending start, stop and parity bits. Here, while the transmit register is transmitting information the transmit buffer register can be loaded. When the transmitter has transmitted the entire contents of the transmit register, the contents of the buffer register are transferred to the transmit register and transmission begins. If the buffer register is empty, the Transmit Register Empty (TRE) output goes "high".

The Buffer Register is loaded by presenting data at the TBR 1-8 pins and then pulsing $\overline{\text{TBRL}}$(Transmit Buffer Register Load) low. TBRE (Transmit Buffer Register Empty) then goes low to acknowledge this. TBRE stays low until the data has been transferred to the transmit register, when it goes high again.

Other lines are control inputs (CLS1, CLS2, PI, EPE and SBS). These select the mode in which the UART operation - see the note on the circuit diagram. Note that to load controls, the CRL line has to be held high. MR is a reset line to ensure the UART is in a known state at power up.

The clock inputs (TRC and RRL) define the Baud Rate of the communcations. The Baud rate is quite simply equal to the number of bits per second transmitted, and the clock input is 16 x the baud rate.

Baud Rate Generation

Several methods of generating a Baud Rate are available. The simplest is to use a timer on the VIA - Timer 1 is the only timer on the VIA suitable. To do this connect pin 40 of the UART to pin PB7 on the VIA (63 on the external bus connector on the Atom) and perform the following operation :

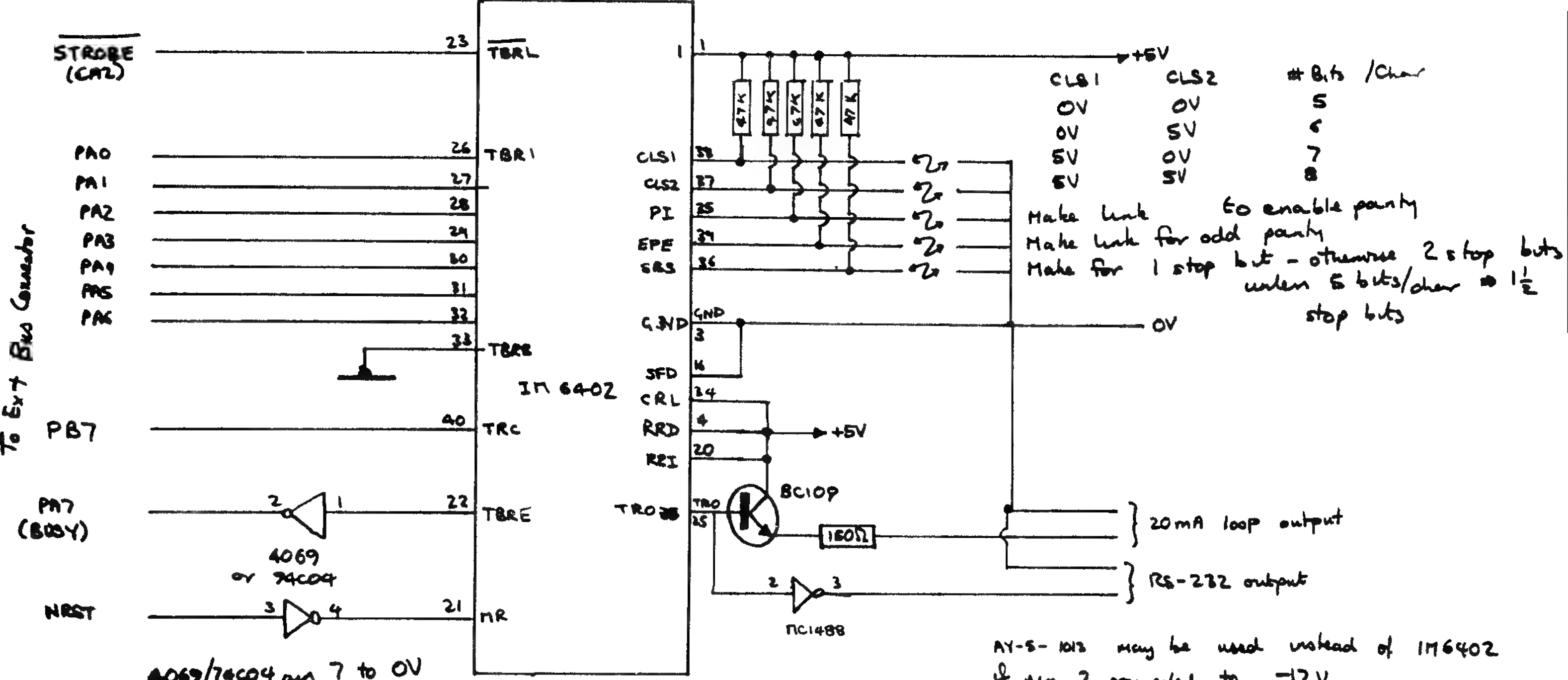
```
? # B8ØB  =  # CØ  init VIA ACR for use of Timer 1
? # B8Ø5  =  )
                   Set Baud Rate
? # B8Ø6 =   )
```

? B8Ø5 and B8Ø6 must be set to $\frac{1000\ 000}{32\text{x Baud Rate}}$

(low byte in B806, high byte in B805) The top 2 bits of B805 must be zero. For an 110 baud teletype these values are B805 = 1 B806 = 28 (decimal). The factor of 32 comes from the fact that 16x baud rate is desired and the output at PB7 changes state every time the counter decrements to zero.

Remember that location FE in the Atom contains a character that will not be sent to the printer. This is initialised to a line feed. If your printer starts printing over what it has just printed, change FE to something else (eg O).

Alternatively, a crystal controlled generator can be used or a 555 - type timer.



CLS1	CLS2	# Bits /Char
0V	0V	5
0V	5V	6
5V	0V	7
5V	5V	8

Make link to enable parity
Make link for odd parity
Make for 1 stop bit - otherwise 2 stop bits unless 5 bits/char ⇒ 1½ stop bits

AY-5-1013 may be used instead of IM6402 if pin 2 connected to -12V.

4069/74C04 pin 7 to 0V
" " " 14 " 5V
MC1488 pin 7 to 0V
" " 1 to -12V
" " 14 to +12V

Interfacing to a Printer

First decide what your printer interface is. Then select your number of bits/characters, number of stop bits, parity and your Baud Rate.

Two common devices are	Format	Interface
Teletype 33 (KSR33, ASR 33, 33RO, DD390 etc)	1 start bit 8 data bits 2 stop bits No parity	20mA loop (note some are 60mA loop or RS-232) 110 baud
Creed teleprinter (75 or 7B) (Note Not ASC11 - Baudot coded	1 start bit 5 data bits 1½ stop bits	High voltage 50/75 baud

Select the format by wire links as shown on the circuit diagram, and connect the correct interface between the UART and the printer (RS232 and 20mA loop are shown).

LOCAL USER GROUPS

Membership of the user group has been rapidly growing recently, to the extent that it is now feasible to run local groups in some areas, especially in London and the Home Counties. Would anyone, in any region, who is interested in running a local user group please contact me. You should find enclosed a membership list for your particular area, which should give you some idea of whether it is worth setting up a local group.

MEMBERSHIP LISTS

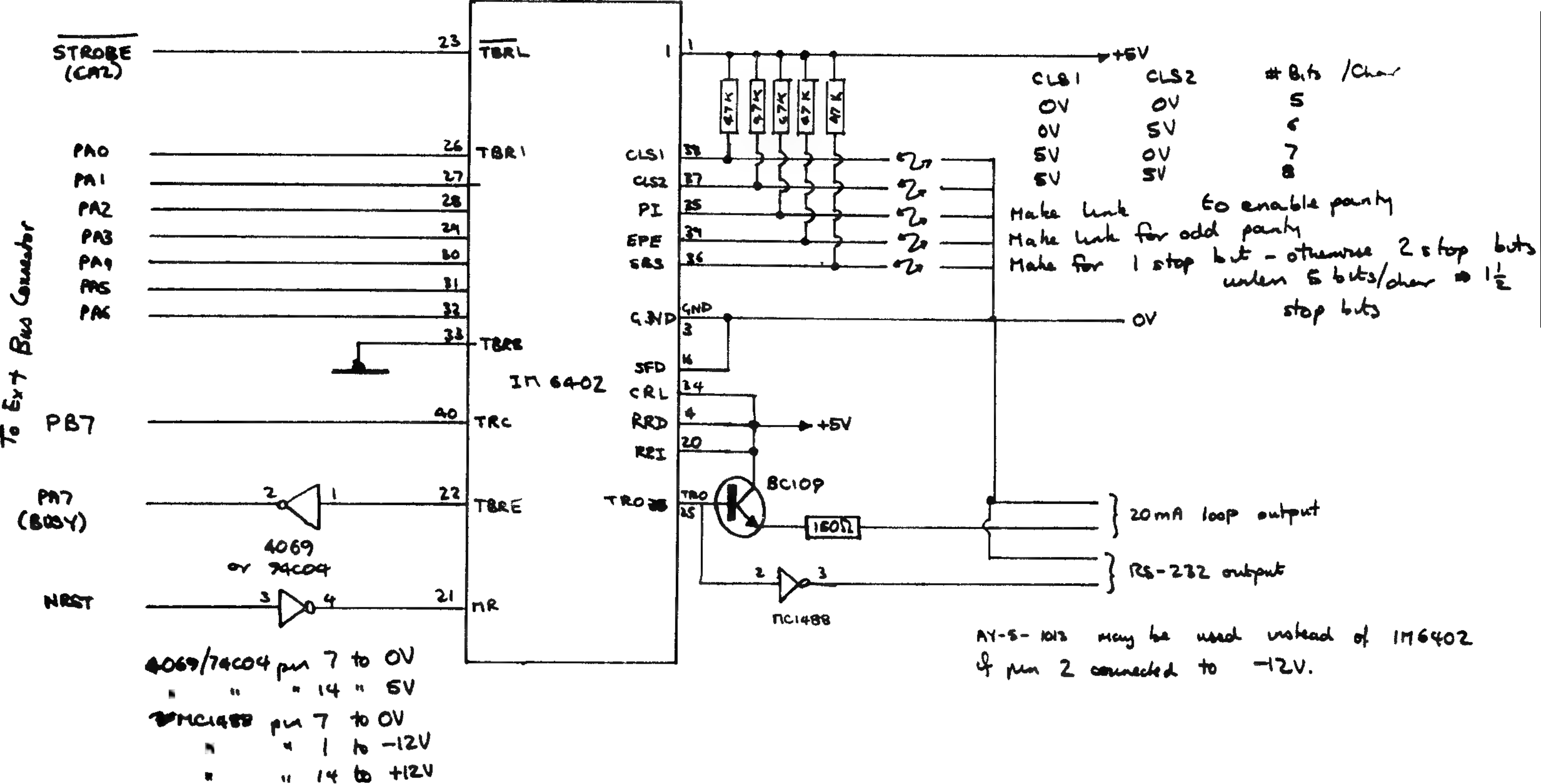
A membership list for your area should be enclosed with this issue. If you live near a county boundary and would like the list for other areas, please let me know.

Alternatively, I can provide a copy of the complete list for the whole world (!!) - although please send 50p to cover Xerox and postage for this.

SECOND HAND

Syd Poole of 7, Islington Close, NEWPORT, Salop, has an Atom power supply he is willing to sell to a member for £7.50 plus post.

If any other members have any computer 'kit' they would like to sell please let me know and I'll put a note in the Newsletter. Remember though the Newsletter doesn't appear every day so there will be a fair delay before the ad actually appears.



CLS1	CLS2	# Bits /Char
0V	0V	5
0V	5V	6
5V	0V	7
5V	5V	8

Make link to enable parity

Make link for odd parity

Make for 1 stop bit – otherwise 2 stop bits unless 5 bits/char ⇒ 1½ stop bits

AY-5-1013 may be used instead of IM6402 if pin 2 connected to −12V.

4069/74C04 pin 7 to 0V
" " " 14 " 5V
MC1488 pin 7 to 0V
" " 1 to −12V
" " 14 to +12V

Interfacing to a Printer

First decide what your printer interface is. Then select your number of bits/characters, number of stop bits, parity and your Baud Rate.

Two common devices are	Format	Interface
Teletype 33 (KSR33, ASR 33, 33RO, DD390 etc)	1 start bit 8 data bits 2 stop bits No parity	20mA loop (note some are 60mA loop or RS-232) 110 baud
Creed teleprinter (75 or 7B) (Note Not ASC11 - Baudot coded	1 start bit 5 data bits 1½ stop bits	High voltage 50/75 baud

Select the format by wire links as shown on the circuit diagram, and connect the correct interface between the UART and the printer (RS232 and 20mA loop are shown).

LOCAL USER GROUPS

Membership of the user group has been rapidly growing recently, to the extent that it is now feasible to run local groups in some areas, especially in London and the Home Counties. Would anyone, in any region, who is interested in running a local user group please contact me. You should find enclosed a membership list for your particular area, which should give you some idea of whether it is worth setting up a local group.

MEMBERSHIP LISTS

A membership list for your area should be enclosed with this issue. If you live near a county boundary and would like the list for other areas, please let me know.

Alternatively, I can provide a copy of the complete list for the whole world (!!) - although please send 50p to cover Xerox and postage for this.

SECOND HAND

Syd Poole of 7, Islington Close, NEWPORT, Salop, has an Atom power supply he is willing to sell to a member for £7.50 plus post.

If any other members have any computer 'kit' they would like to sell please let me know and I'll put a note in the Newsletter. Remember though the Newsletter doesn't appear every day so there will be a fair delay before the ad actually appears.

ATOM USER GROUP

Newsletter 3   May 1981

Firstly, apologies to all of you who have waited so long for this copy of the Newsletter. This was due to production diffficulties which have, now, hopefully been resolved. I would like to assure all members that their subscriptions will cover four issues of the newsletter.

Many thanks for your comments about the first two issues. as you can see, there have been only minor alterations such as allowing space for a left margin for those who wish to keep their copies in a binder. Although such details may seem trivial, they are important - please tell us of any other suggestions that you may have. Several people have complained about poor service from the group in the past. While I believe that such cases are not too widespread, they have in many cases been justified. Therefore, the software library is now being dealt with by Peter Frost, to whom all correspondence concerning the software library should be addressed. All other correspondence should be addressed to me. We hope to give a better service in the future. To all those who have suffered delays, our apologies.

The address which should be used in the future for all correspondence, whether to Peter or myself, is:

18 Frankwell Drive,
Potters Green,
Coventry.

Contents of this issue:-

Contributions

Contributions for the Newsletter or software library are always welcome, and there has been a good response to both.

However, would people please observe the following guidelines.

Software

Please will you send, as well as the programme, a note giving what it does and instructions of how to use it? You may know how it works, but we do not necessarily, and nor do we have the time to find out.

Please will you also, if at all possible, send a cassette as it is impossible for us to type it all in to get a neat printout and master cassette for distribution. If possible, a listing would also be appreciated if you have a printer.

Articles

While there is plenty of material coming in, there are relatively few coming in a form which can be directly included without a lot of editing. For an example of an ideal format, see Tim Jackson's articles in both this issue and Newsletter 2, neither of which have been altered at all other than retyped.

Hardware Library

There has been a low response to this in terms of contributions received, although there have been requests. For this work there must be some stuff coming in, otherwise it will have to be discontinued.

The essential qualifications for material for this are:-

(a) The original drawing must be in a form suitable for reproducing, by printing or photocopying.

Black is best for copying, followed by red. Blue is poor, green will not copy at all. The absolute ideal is indian ink on plain white paper but a black biro or fibre tip (use a fine one) will do very well.

(b) Information must be sufficiently complete that it is possible to build and use the device successfully - i.e. give connections, power supply voltages etc.

Bits and Pieces

Information on digital waveform generation and synthesis can be obtained from: Paul Beverley, Department of Electronics, Norwich City College, Ipswich Road, Norwich Tel. 0603 60011 extension 231 or 233 (day)
or 0603 610622 (evenings)

Time Trials Paul has also sent me a long list of time trials he has conducted on the Atom, finding out the time the Atom takes to perform its various functions. If anyone would be interested in a copy, please let me know (normal copying charge will apply - see elsewhere in this issue).

Interest Groups I am trying to compile address lists of members with similar interests. At the moment, I only have a list of Radio Amateurs, complied by going through the address list picking callsigns out! If amateurs other than G8LBM, G4BZU, G3NFL, G4GLM, SM5 AKS and SM6 GXV would like their names and/or callsigns on the list please let me know.

It is rather harder for me to pick out people interested in educational applications, which seems to be another worthwhile area to compile a list, due to the fact that a school or college address may or may not mean anything. Would then anyone interested in being on such a list please contact me.

If anyone would be interested in appearing on a list of any other interests, please let me know and I will try to compile one if there is sufficient interest.

Jump Indirect I am informed, although I have not checked it, that there is an error in all 6502 processors. This is not specific to Atom, but will occur on all machines that use the 6502 (Apple, Pet etc.)

If the first byte of the address is on the last byte of a page, the second byte will be obtained from the first byte of the same page rather than the next page.

For instance:- JMP (#31FF) The second byte is fetched from #31ØØ rather than #32ØØ.

Cassette Recorders

A number of people have encountered problems while saving and loading tapes. Due to the wide variety of tapes and tape recorders available, it is not easy to diagnose problems at long distance (or short distance very often). The following article, contributed by R. Bostock, gives some of his observations about the problem.

One observation he does not make in the article, but did mention in his covering letter, is the importance of decent quality tape. Our ears can filter out imperfections in the sound produced by a tape recorder but the computer isn't as clever. It is essential to use a reasonable quality tape. The cheapest nastiest tape is, quite simply, useless for computer application. Due to my considerable tape usage, I buy specially wound tapes which are not available through normal channels but commercial brands which are good are TDK (D or AD), Maxell UD and BASF SM. These do cost more but they are considerably more reliable than the cheaper brands.

A very wide range of recorders can be used. I use a stereo Hi-Fi deck (as I had it and didn't want to get another player) while the cheapest DC has DC erase auto level control etc. machines can also frequently be used successfully.

Now over to Mr. Bostock:-

CASSETTE RECORDING - RECORDERS

One of the problems besetting Atom (and other) computer users is the incompatibility of cassette recorders, and thus the difficulty of loading a programma tape made on a recorder other than one's own. Some investigations into this using several recorders produced some significant findings.

Speed

Absolute speed is not so critical as might be expected, even a small amount of wow (slow speed variation) can be accepted. Flutter or high speed variation less so. Many of us use cheap machines on which exact tape speed may not be provided, in normal use with one Atom and its associated recorder working as a pair recording and replaying at the same speed no problem arises.

Motor Noise

Some degree of motor noise is often present during record or replay. This can usually be dealt with to some extent by fitting a 0.01 ufd and a 1000 ufd electrolytic capacitor in parallel across the motor feed, as close to the motor as possible. The electrolytic must be connected with the correct polarity of course.

Head Alignment

This was found to be the major culprit, the Atom is happiest with equal amplitude of the 2.4Khz and 1.2Khz input signals. If the record/replay head is incorrectly aligned the 2.4Khz signal will be lower than the 1.2Khz. Some assessment can be made by listening to a replayed tape; the high tone leader should be clear and strong and equal to the following mixed tones. Any wow or flutter will be heard as a variation of the high tone leader. A new drive belt often helps this problem.

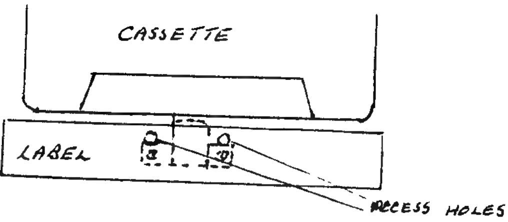
Between the polefaces of the head is a gap (vertical) of less than .001", see drawing, for best high frequency response the gap must be exactly perpendicular to the run of the tape providing the shortest possible path across this gap. If the high frequency response is good the low frequency response will be good. The converse is not necessarily true. Machines vary as to the access to the alignment screw some needing to be removed from the case entirely. Most have an access hole covered by a name plate or instruction plate immediately over the head, this can be eased up if it is a thin aluminium sheet glued on, or if a plastic plate it should unclip in some way. The head is mounted on a pillar at one side by a single spring leaf, the alignment screw on the other side goes into the chassis through a coil spring which supports the head. Both screws are usually sealed with a dab of paint.

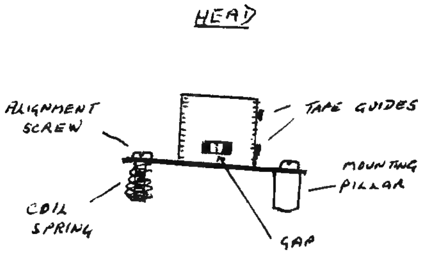
To align the head first gain access to the alignment screw by removing the label or panel. Make a cassette recording with a long high tone leader from the Atom and replay it on the recorder listening to it on the speaker. With a fine screwdriver adjust the alignment screw for the loudest and best response from the high tone, this response will be sharply defined. When this has been achieved re-record the high tone from the Atom and again replay it as a check. Lock the screws with a dab of cellulose paint or some such quick drying paint and allow to dry.

Try loading a programme, if improvement has been made a lower volume level may be required.

If the recorder is fitted with a DIN input/output socket, it is worthwhile trying this. This output is taken off before the volume control and any tone correction on the machine. It leaves one with a volume controlled speaker output for audible monitoring whilst a programme is being loaded.

In conclusion a good head cleaning tape, regularly used, and choice of good quality cassettes is an essential.





Hex Dumps

A few people have asked about loading Hex dumps into the Atom. The Atom lacks a machine code monitor of the type many machines have, however such a monitor is available from the library (DEBUG).

Frequently though all that is required is that a dump be loaded into memory, and the more sophisticated features of a monitor are not required. An example is the Hex dump some software library programmes are supplied with.

A short routine for loading such dumps is given below. It is fairly self explanatory but it should be noted that a hash is (#) is required to signify Hex as a reply to the EXAMINE FROM? Prompt. Elsewhere, Hex is assumed. To leave a byte unchanged, type RETURN only. To change it, type the new value followed by RETURN. Exit by typing ESCape.

```
  5 DIM B64;@=4
 10 INPUT "EXAMINE FROM "A
 20 DO
 30   P.&A,&?A                         45  L = LENB
 40   IN. $B
 50   ~~IF LENB<>0 GOS.a~~  IF L<>0 ; GOS.A
 60   A=A+1
 70 U.0
 80aX=0
 90 FOR Z=0 TO 1
100   B?Z=B?Z-#30
110   IF B?Z>9 B?Z=B?Z-7
115   IF B?Z>15 P."ERROR"';Z=1;N.Z;R.
120   X=B?Z+X*16
130   ~~IF LENB=1 Z=1~~  IF L=1 ;Z=1
140 NEXT Z
145 ?A=X
150 R.
```

TV Picture

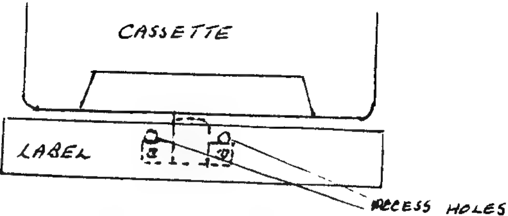
I have received a letter from Acorn about a problem they have encountered with the TV interface, of which I reproduce the text here.

Dear Mr. Meredith,

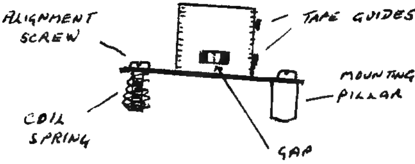
We have recently had several enquiries concerning problems with the Atom, in particular inability to hold a picture on the screen. These have often been caused by power loss in the lead we were supplying with the Atom PSU, which is about 1 - 2 yards long. The problem is solved by cutting the lead down to as short as possible, which we have done with our newer PSU's.

Could you draw this to the attention of the User Group through your newsletter, so that others who may be suffering from this can take suitable action? The power loss has been as much as 0.8 volts in some cases.

I hope that this clears up some troubles!


CASSETTE
LABEL
RECESS HOLES

HEAD


ALIGNMENT SCREW
TAPE GUIDES
MOUNTING PILLAR
COIL SPRING
GAP

RAM Board

Lloyd Nolan, of 99 Fir Street, Cadishead, Manchester, M30 5AR is considering producing a circuit board for a RAM card for the Atom. He envisages it would be of about 16-17 K capacity, and would cost around £70. If anyone is interested, please contact him at the above address or on 061 775 0209 after 6.30 p.m. He asks me to make it clear that at the moment he is really only interested in numbers and cannot give any replies unless specifically requested.

Bugs

Two bugs have recently surfaced in the Acorn BASIC ROM in the Atom, which it may be useful to be aware of:-

(1) In the assembler, the array AA cannot be used for labels. If this is tried, an error results, for instance:-

```
:AA3  LDA   @#40   This is okay
      JSR   #FFF4
      JMP   AA3    but this gives an error!
```

The cure for this is not to use the array AA for labels in the assembler.

(2) Also in the assembler, the use of the CH function as an operand causes a defective assembly listing to be produced. Specifically, on the listing produced when the programme is RUN the line containing the CH function is not printed up to the closing quote, i.e.

```
LDA   @CH"@"   This is a comment
```

is printed on the listing as:-

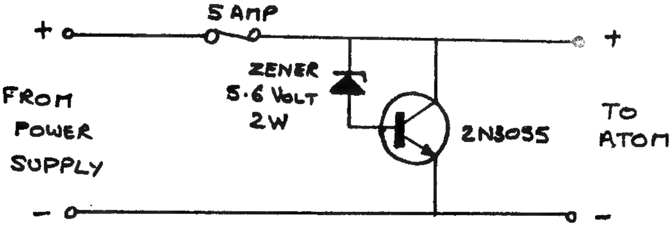
```
"This is a comment
```

The code produced however is correct.

Power Supply

Further to the power supply design published in the last issue, I have been told that it is not unknown for 78H05 regulators to blow up, shorting the input and output together and thus applying 12V or so to the Atom board - and blow up practically everything on it.

This can be overcome by fitting a "crowbar" circuit to the output of the supply as follows:-



If the output of the supply ever goes above about 5.5V the zener diode will conduct heavily and turn on the transistor which will blow the fuse, thus protecting the Atom. A 400 mW Zener can be used up to about 2A, above this (to about 5A) use a 1W Zener, above this use a 10W Zener.

This circuit has the further advantage that transients are removed which may otherwise cause the Atom to "crash" (for instance if the freezer thermostat switches on!)

Many thanks to Kamal Jabbour of Manchester for pointing thus out.

Disclaimer

The Atom User Group provides all information in the belief that it is correct. No responsibility can be taken for any loss or damage, real or consequential, for the use or abuse of any such information.

Useful Addresses

Software Delays can be obtained in assembly language by using the following calls:

```
JSR   #FB7D    2 seconds
JSR   #FB81    ½ seconds
JSR   #FB8A    1/10 seconds
JSR   #FB83    X/60 seconds
```

These may also be used with the Basic LINK command. Register X is used.

The contents of A can be converted from ASC11 to binary by JSR #F87E. On return, carry is set as follows:

Carry Clear if character in A was Ø - 9 or A - F. Carry Set otherwise.

JSR #F7F1 prints the 16 bit number at (X), (X + 1) in zero page in Hex.

For instance:

```
LDX  @2
JSR  #F7F1
```

will print the contents of locations 2 and 3.

JSR #F893. This reads in a Hex number from the BASIC line buffer at 100 upwards, starting at the Yth character. The number read is placed, in binary, in locations (X) and (X + 1). At exit, Y points to the character encountered which was not a valid Hex character. A is destroyed and (X + 3) is used.

JSR #F876 This searches for the next non blank character in the line buffer starting at the Yth character.

(224) If this byte is negative the screen is 'off' i.e. equivalent to P.$21.
If positive screen is 'on', i.e. P.$6

(234) If bit 2 is set, Tape messages are disabled (i.e. * NOMON)
If bit 2 reset, they are enabled (* MON)

The addresses of the start of the arrays @@ - ZZ are stored in the following locations:-

@@	747	and	774
AA	748	and	775
BB	749	and	776
		thru	
ZZ	773	and	800

All decimal
not hex addresses.

Running Basic programmes with *RUN (see Newsletter 2)

I said in the last issue that BASIC programmes could be loaded and run from tape automatically by use of the *RUN command if they were stored with an entry point of CE86 instead of C2B2. I also said that this didn't work if you dimensioned any arrays.

Apparently this is because that *RUN, unlike RUN, does not setup the free space pointer (?35and ?36) or TOP, and hence anything that uses these will not work. Thus the free space pointer needs to be explicitly set before executing any DIM statements.

This can be done by inserting the following line at the start of any programme which uses *RUN and arrays:

```
1   ?35 = Ø, ?36 = #84
```

which will place arrays at #8400 on. Of course, any available free memory may be used.

Note that !35 = #8200 will also set 37 and 38 to zero which may cause mysterious results and !35 = TOP won't work as TOP isn't set either. TOP is located at 13 and 14 decimal. See Newsletter 2

Small Ads

FOR SALE - ASR33 Teletype, papertape reader/punch, everything works. Complete with 2 rolls papertape and plain paper. 20mA loop. Sensible offers to J. H. FORD, 41 EDWARD AVENUE, CAMBERLEY, SURREY. Telephone 0276 22585.

2114L RAM chips suitable for Atom £4.50 per pair (1K) including post.
A. K. BOURNES, 26 HIGH STREET, WOULDHAM, ROCHESTER, KENT.

"Relocating Arrays", and the note on *RUN ing programme in Basic.

Point Plotting Routines

The point plotting routines used by Basic are also usable by assembly language programme. See page 88 of AT & P for details of passing parameters etc.

However, the BASIC CLEAR statement will have to be used to put the correct value in the vector at #3FE.

To avoid this, the following routines can be used:

#F6E2	Mode Ø
#F73B	Mode 1
#F754	Mode 2
#F76D	Mode 3
#F7AA	Mode 4

These will, of course, require that the screen be in the correct mode by setting #BØØØ to the correc value first.

New Charges

Unfortunately, due to the continuation of inflation and the recent postal rises it has been necessary to introduce some charges and increase others. Charges as from now will be:-

Photocopying/Duplicating — Please allow 15p per item to cover copying. This charge includes address lists, documentation and listings, plus any other items that may become available.

Back Issues — Price of these remains unchanged at 50 pence per copy.

World Address List — These are now £7.50 a copy - not 50 pence as stated in the newsletter!

Cassettes — These will now be changed at £1.00 per cassette. These, as before, are C10 or occasionally C20.

Please note that, due to the amount of duplicating we need to do, it is not possible to provide software on a user-supplied cassette nor is it possible to provide more than one programme per cassette.

There are two exceptions to these charges, which include postage.

If the total money due comes to less than 50p, then no money will be due. I am quite aware that this is open to abuse, however if extensive abuse does take place, the concession will be stopped.

The second exception is that a cassette containing any contribution to the library is sent, rather than return you original cassette (which may be subject to delay) it is possible to return, free of charg a cassette from the software library. Please state, when sending such a cassette, if you would like to take advantage of this.

Apologies are made for these increases, however they are necessary to keep up with inflation.

IMPORTANT

Cheques in payment for anything the User Group provides should be made payable to the "Atom User Group". They should not be made payable to Peter Frost or Richard Meredith, or any of the more exotic names I have received.

Despite Personal Computer World's attempt to change my name, I am still R. G. Meredith and not T. G. Meredith.

FOR SALE - Quantity of components as follows:-

MONSANTO MCT6 dual opto coupler 4 for £1 including data.

TTL etc. - 74LS151 40p, 74116 75p, 74L93 40p, 74L04 10p, 4011 10p, 4025 10p

RS Transistor Relays 12V supply, 2.3V 700 mA operating current, heavy duty contacts (about 10A) £1.50 each (allow 50p minimum for post). Few only of these.

Please allow 30p for post - munimum order £1 please! All believed unused and to full specification.
R. MEREDITH, 18 FRANKWELL DRIVE, POTTERS GREEN, COVENTRY. Cheques payable to R. G. Meredith please not the group!

Keyboard Input Routines Addendum

The article on Keyboard Input Routines in the last issue attracted a lot of correspondence. However, Tim Jackson, who wrote the original article, has been ferreting around further and his next article is included below. This should satisfy practially everyone now.

To all those others who sent information in - many thanks; the only reason I have not included it is that Tim's solution appears so comprehensive.

One comment however that is worth making is that, when fiddling round with port #BØØØ in assembler, if the low 4 bits are not reset to zero before exiting, the program may exit prematurely if any column 5 keys are pressed, as though ESC were pressed. The cure is simply to make sure you do reset column 5 to zero!

Keyboard Input Routines - Postscript.

Tim Jackson

Since writing Keyboard Input Rountines in the last issue, I have been investigating the workings of the operating system OSRDCH function, which starts at #FE94. It makes use of a subroutine at #FE71, which performs a single keyboard scan. If this subroutine finds a key is being pressed, it clears the carry flag and exits with a value between 0 and 59 (decimal) in the Y register, depending on the key. If no key is being pressed the carry flag is set on exit, and Y contains #FF.

The value in Y is not the ASC11 code of the pressed key - it depends on the position of the key in the keyboard matrix shown in the Atom's circuit diagram. ESC is 59, Z is 58, Y is 57, and so on down to [= 1 and space = 0. SHIFT, CTRL and REPT are ignored.

This can be used in real-time graphics programmes by calling a routine such as:

```
JSR   FE71
STY   80
RTS
```

and then checking ?#80 for the value of the desired key or keys.

If you want the ASC11 value of the pressed key without calling OSRDCH (which waits until a key is pressed) try the following:

```
: LL0   JSR  #FE71
        BCS   LL1
        JSR   LL2
: LL1   STA  #80
        RTS
: LL2   PHP
        CLD
        JMP  #FEBI
```

(A two-pass assembly is required. Call using LINK LL0)

The JMP #FEB1 instruction jumps to the middle of OSRDCH, where there is a decoding routine. JSR LL2 is balanced by an RTS at the end of OSRDCH. The PHP is necessary to balance a PLP at the end of OSRDCH.

The net result of this is that a single keyboard scan is performed, and if a key is found to be pressed its ASC11 code is stored in 80. If no key is being pressed, zero is stored in #80. (Note that ESC and CTRL - @ will also give zero).

SHIFT and CTRL have their usual effects, and ATOM control codes can be generated (though they will not be acted on). The result can of course be stored elsewhere if desired, by changing the STA #80 instruction - for example STA #322 puts the result in the lowest significant byte of BASIC variable A.

This routine is a distinct improvement over the BASIC keyboard scan given in the last issue, since it is possible to detect keys R to Z, and to use SHIFT and CONTROL. The only keys it does not give satisfactory results with are LOCK and the cursor control keys (the simple JSR #FE71 routine does work with these keys). ESC, of course, is likely to give problems when you return to BASIC.

The Back Page

Yet another set of errors again I'm afraid:

ch. 14.2	Don't use the array AA for labels due to the bug in the Assember (see "Bugs").
User Group Library DIFFEQ	All listings supplied so far contain an error. Line 175 FOR T=Ø TO 255 should be inserted.
AT&P Square root programme p35	The problem mentioned in the last newsletter (arithmetic overflow) can be "fixed" by using a "saturating" register, i.e. by inserting:- 115 IF Q>65535 THEN Q = 65535
AT&P p193 line 9	Should read #FFEØ OSLOAD JMP (LODVEC) not #FFED
AT&P p194	#7Ø - #7F should read 112-127 not 112 - 143 #8Ø - #AF " " 128-175 not 144 - 175
AT&P p172 23.3 and Newsletter 2	Don't put 5 DIM P11 as per last newsletter. Change line 10 to 10 DIM ZZZ, P-1 (P must be last variable dimensioned, see p171).
AT&P p37 & 36	Not really a "bug" but:- To prevent cheating with the Reaction Turner by continuously tapping the key, insert the line :- 35 IF ?#BØØ1 <> #FF PRINT "CHEAT!"'; GTO 120
Newsletter 2	*DOS may or may not cause an error to be signalled. It may however cause the ATOM to 'hang' and require BREAK to reset it.
Atom Technical Manual.	When wiring up the extension socket, side a and side b have been transposed. Lines a1 to a32 are really b1 to b32 and vice versa.
AT&P p55 7.3.3	Line 5 should be 5 T=#37FF not 5 T=#3BFF Lines 95,100 and 110 should be 95 IF L!P < 100 P.Ø 1ØØ IF L!P < 10 P.Ø 11Ø IF L!P < 1 P.Ø

Local Groups

Since the last newsletter, I have had news of the following local groups of Atom owners:-

Norwich

Paul Beverley, Department of Electronics,
Norwich City College,
Ipswich Road,
Tel. 0603 60011 extension 231 or 233 (day)
0603 610622 (evenings)

Manchester

Clement M. Rutter,
3 Leopold Avenue,
Withington,
Manchester. M30 8JG
Tel. 061 434 3092

London

John Berry,
64 Palermo Road,
London. NW10 5YP
Tel. 01 961 5148

SOFTWARE LIBRARY

First of all many thanks to all of you who have submitted software to the library, the quality of some programmes is most encouraging. When dividing what to finally include in the library I shall try to cater for all sizes of Atom. Most of the programmes received so far have been suitable only for expanded Atoms, so if you have a minimum machine your contribution would still be most welcome.

Naturally games constitute the majority of software submitted, with some utilities but very few educational or scientific programmes so there is certainly plenty of scope in these areas. Don't forget that the Atom does have a very good assembler and it is a pity not to see it being used as often as it should.

Many apologies for the apparently long delay in replying to orders for software since January. This has in main been due to the huge backlog generated over Christmas, a shortage of blank cassettes and the change over period when Richard handed over the software library to me. Hopefully from now on software requests will be fulfilled within 21 days of receiving the order.

A couple of points concerning ordering items from the library. It would help tremendously if, when ordering listings if they are listed in alphabetical order, also all cheques should be made payable to the Atom User Group, not to me!

Lastly, please submit all software on cassette as the time involved in typing in listings is prohibitive.

Many thanks. Peter Frost,

18 Frankwell Drive
Potters Green
Coventry CV2 2FB.

Programmes currently available are:-

UTILITIES

RENUMBER

A fast renumber routine written is assembly language suitable for all sizes of Atom.

PACK

A machine language utility for reducing the size of a programme by condensing multiple spaces into a single space. This method ensures that syntax errors due to missing sifnificant space cannot occur, and readability is improved although the space saving is less than that obtained by eliminating all spaces.

Please note that although both programmes will run on an unexpanded Atom (both run in screen memory) the programme will need assembling in stages to run in a 2K Atom, as the source is of over 512 bytes. No problems should be encountered in Atoms of over 2K.

HWRITE

Allows mixing of high resolution graphics (mode 4) and text, by replacing the existing VDU routines with another which creates the dot pattern needed to form 64 character ASCII subset, as well as recognising most of the Atom's control codes.

Only really suitable for 12K Atoms, as the whole of the upper text space is used by the graphics and the binary code takes 1K. Assembler.

DEBUG

A debugger for machine code programmes similar to the monitor found on machines like the Acorn System One and the Nascom-1. Written in a hybrid of Basic and assembler. Size 3105 bytes source, 30 bytes machine code. No additional hardware needed.

AIR RAID

To land your plane you have to bomb the runway clear first, Airport '82 perhaps.

TEXT 1024
VDU 512

BIO

Find out just when you are at your intellectual physical and emotional peaks.

TEXT 4K
VDU 6K

BOMBS AWAY

Like AIR RAID but uses high resolution graphics to generate the profile of a city.

TEXT 4K
VDU 6K

BREAKOUT

Demolish a wall of bricks with a bat and ball, with sound effects.

TEXT 3½K

DODGEMS

Navigate your way through a maze collecting points on the way while being chased.

TEXT 3K
VDU Mode 3

DOMINOES

Play against the Atom with this version of 5's and 3's.

TEXT 4½K
VDU 6K

DUCKSHOOT

You have just disturbed a field of ducks and have to shoot them before they fly out of range.

FIND SNOOPY

Find Snoopy lost in a forest using only compass points to direct you.

TEXT 2K

LUNAR LANDER

Land your space craft on a randomly generated landscape which enlarges as you get closer.

TEXT 3K
VDU 6K

MAZE

Navigate your way out of a maze using 3D projection of its passages to help you..

TEXT 4K
VDU 6K

MISSILE

Use skill and judgement to eliminate passing ICBM's.

TEXT 1K
VDU 512 BYTES

MOONLANDER

Given a certain amount of fuel you have to land your spacecraft gently on the moon.

TEXT 1½K

MOUSE TRAP

Catch a mouse by "drawing" a trap around it as it moves over the screen.

TEXT 1.5K
VDU MODE 1

OTHELLO

Play against the Atom or a friend using this fast response time game of othello.

TEXT 4½K
VDU Graphics 3

PASSE-TEMP

Play against the Atom in this version of "connect 4".

TEXT 3K

PONTOON

An Atom version of this popular game.

TEXT 3K
VDU 1K

SHARP

One for those with elephantine memories. You have to remember some numbers that have been displayed for a short time. But as you get better it gets harder.

VDU 512 BYTES
TEXT 1K

SPEED BOAT

Navigate your way round a lake avoiding various obstacles using the appropriate nautical terms.

TEXT 4K

SUBHUNT

Game of "battleships"

TEXT 512 bytes
VDU 512 bytes

TARGETS

A Target is displayed on the screen and you get 10 shots to shoot is down.

TEXT 512 bytes
VDU 512 bytes

EDUCATION

FRENCH VOCAB

Test the vocabulary of a French student. Not available on cassette.

MENTAL ARITHMETIC

Sets a child some sums to do and check his or her answers. Not available on cassette.

OTHERS

ETCHA

Allows you to "draw" pictures on the screen (like the Etcha-Sketch toy). Works in graphics Ø.

TEXT 512 bytes
VDU 512 bytes

SPIROGRA

Draws Spirograph patterns in graphics mode 4. Requires full 6K screen memory, 1K lower text space and floating point ROM.

DEFFEQ

Solves second order linear differential equations and plots their response to a step input in graphics mode 4. Requirements as SPIROGRA.

SERIAL

I/F Serial Teletype/VDU driver to replace the inbuilt memory mapped display of the Atom. Written in assembler, it is suitable for all Atoms of over 4K which have at least 4K available in a continuous block. (for the source code). The binary code is less than 256 bytes.

Any Atom with the lower text space in operation can run the binary. Either the display only or the keyboard and display can be replaced. Requires the 6522 VIA and a modicum of external circuitry to drive either a RS232 or loop terminal.

SWUART

The software UART routine used to provide serial i/o in the above. As above, but no control code recognition or operating system interface. Hardware requirements 3K RAM, VIA, minimal external circuitry.

CONFACTORS

A programme for giving approximation of required ratios for gearing in machine tools and pairs of factors to help in selecting suitable pick-off gears. Not available on cassette.

Atom User Group
Newsletter 4 Sept 1981

The Atom User Group is now one year old, and we must now take the opportunity to thank those of you who have given us so much support for the past year of our activities. We now number some 2000 members and this number is still steadily increasing.

In the next few months however there will be some fairly substantial changes in the User Group. I am having to give up my part in the User Group as I am going into business as the Technical Director of a company specialising in production control equipment in Cambridge, and obviously my commitments there will prevent me from having an active part in the User Group. Peter Frost, who has until now been running the software library, has agreed to take over my duties and so all future correspondence should be addressed to him.

One or two people who have in the past contributed have written in to ask why whatever it was they contributed has not appeared. With 2000 members we get so much sent in that some contributions are squeezed out through lack of room. Please don't be discouraged if something you've sent in doesn't appear - its all appreciated - honest!

We have also been asked if we can review some of the software availiable for the Atom. With this in mind, we include reviews of a couple of pieces now availiable for the Atom.

Lastly, the Atom User Group will be represented at the Personal Computer World Show, to be held at the Cunard Hotel, Hammersmith, London on the 10 - 12 September. We hope to see you there.

RICHARD MEREDITH.

Contents

Willow Software Utility ROM

The Utility ROM provided by Willow Software is, as its name suggests, a set of utilities for the Atom and I would imagine it is similar to the Toolkit ROMs availiable for the PET. It is installed in the extra ROM socket on the Atom PCB. And here was the first problem - on my Atom, that socket is already occupied by the Acorn Word Pack. Thus a certain amount of opening the back is required... I would advise anyone who may be intending to use both to fit a zero force socket in that position. It may save you breaking the lead of an expensive part or wrecking a cheap IC socket and damaging the Atom PCB when replacing it, which could be very expensive.

The manual is well produced and comprehensive. All the commands are explained in detail and are listed in alphabetical order.

Facilities Availiable

To deal with the extra facilities in more detail, the commands added are ALERT, APPEND, AUTO, BDUMP, BLIP, BSAVE, BZERO, DEL, DIS, FIND, KEY, KRUNCH, MDUMP, REN, SIZE, SPACE AND XREN. In addition, it provides a register dump routine for use by assembly language programmers.

All functions appear as COS commands - ie they must be prefixed with an asterisk.

ALERT bleeps the speaker at the end of all subsequent cassette input and output commands so you know when it's finished. After issuing an ALERT all subsequent loads etc are bleeped until it is turned off with another ALERT.

APPEND loads a BASIC program after the end of the current program, and resets the value of TOP to reflect the size of the new, enlarged program.

AUTO provides an auto line number facility. You can specify the start line number and increment or you can default. If the line already exists, the fact is flagged in case you don't want to change it. If you don't however you have to leave AUTO.

BDUMP prints out the values of all the nonzero integer variables. This can be very useful when debugging programs.

BLIP causes the speaker to be blipped after every key depression. This is intended to be an aid to typing but I tend to hit the keys so hard that I get a pretty convincing click anyway!

BSAVE saves a BASIC program so it can be loaded and run with *RUN - seems we've heard a little about this before! I didn't have time to test it but it claims to have solved the problem of dimensioning arrays.

BZERO sets the values of all the integer BASIC variables to zero.

DEL allows the deletion of a range of line numbers. In the normal Atom, if you want to delete all the lines between 100 and 300 you have to delete them individually. DEL 100,300 will also do this. It is very similar to LIST in that *DEL ,100 will delete lines below 100 etc. One nice touch is that if you type *DEL by itself you don't lose it all (cf LIST!)

DIS is a fairly standard disassembler. My only criticism was that it didn't display the ASCII character as well as the hex data, which can be a great help distingishing text from other data. However MDUMP can be used for this.

FIND prints out all lines which contain a given string, or optionally all lines in a given range. For instance, all FOR statements can be printed out with *FIND/FOR/.

KEY is a solution to another perennial problem - trying to input a character without waiting if no-one's pressing a key. unfortunately, if you press one of the cursor control keys it stops until you press another key.

KRUNCH compresses a program by removing all spaces, except those within strings, and REM statements. It isn't clever though and removes significant spaces such as those in the statement FORA=B TOC STEPD.

MDUMP gives a hex dump, 8 bytes to a line, of a specified area of memory.

REN is a full renumber utility which renumbers GOTOs and GOSUB destinations as well as line numbers. There are a few pathological cases that can't be renumbered automatically and the manual slips slightly here by not mentioning them. An example is the statement GOTO N where N is a variable. Obviously, the renumberer can't predict what value N will have at runtime so it can't renumber it.

SPACE prints out the amount of free space at the end of any program.

XREN is a simplified REN which does not change the destinations of GOTOs and GOSUBs; intended for use with APPEND.

Conclusions

The facilities offered are a good, useful set which should appeal to all programmeres, whether they are interested in Basic or assembler. It is unfortunate you can't use both Word Pack and the utility ROM simultaneously but that isn't the program's fault.

At a price of £35.00 all inclusive it is perhaps a little expensive but it is very useful. I would certainly recommend it.

Small Ads

Adverts are accepted free of charge for this section from private advertisers only. Please contact the person concerned for more information, not us!

FOR SALE - Brand new unused floating point ROM for Atom with National data sheet. £19.50. J Stobseth - Brown, 114 Kirby Rd, Portsmouth, Hants. Tel 61537

FOR SALE - Data Dynamics 390RO Teletype £115.00, Cased Superboard II c/w expansion bd, PSU, software £125.00, Set Elektor SC/MP PCBs, not complete (PROMS need programming etc) £25.00. RW Hearn, 10 Speedwell Close, Pakefield, Lowestoft, Suffolk NR33 7DU.

HELP! Has anyone written a RTTY program for the Atom. If so, I'm interested. J RIGGS G4KDK, 68 Gomer Lane, Gosport, Hants PO12 2QN.

Useful Addresses

Another goodly crop unearthed by users poking about in the Basic:

#F141 This address should be used in preference to #CE86 for self running BASIC programs. This enables the use of arrays etc within such programs provided TOP is set first. This can be done by inserting the following line at the start of the program: 1 !#0D=#0000;!#23=TOP.

After entering this line print out the value of TOP and enter it in place of the dummy value 0000. This ensures that the program does not change its length. The program can then be saved with *SAVE name 2900 **** F141 where **** is the value of TOP which was printed out. The program can then be loaded and run with *RUN.

Another useful idea is to load a very long program in two or more partsand provide a CHAIN function as found in some BASICS. This is possible because we can include *RUN in programs. To CHAIN a program one might make the separate parts self running and chain them using *RUN. Please note I haven't tested it - I'm just throwing the idea out for experimentation.

19
20
21 These three locations contain the depth to which DO UNTIL, GOSUBS and FOR NEXT loops are nested.

224 As well as its use described in newsletter 3, this location contains the position along the line of the cursor. This can be used as a TAB function, ie ?224=20;PRINT"A" is equivalent to the PRINT TAB(20),"A" found in other BASICs.

#FE66 Wait 1/60 second

#FD1A Bleep (as P.$7)

#FE40 If the accumulator is less than 5, the message PLAY TAPE is printed, if equal to 5 REWIND TAPE, and otherwise RECORD TAPE. The Atom then waits for any key to be pressed.

#FE71 Scan keyboard and return 255 if none pressed. Return a unique value if a key is pressed. The value is returned in the Y register.

3 Points to character being interpreted in the current line.

5,6 Start address of statement being interpreted

#DD FLOAD flag. Ignores block nos if bit 7 set.

#E6 Number of lines remaining if VDU in page mode (bit 7 = 1 if not in page mode).

#33D thru #3C0 Addresses of labels. Label a is in #38D and #38E, b in #38F and #340 etc.

#F87E Converts ASCII digit in accumulator to binary; returns carry set if not 0 thru 9.

REVIEW: Acorn Word Pack ROM

The Acorn Word Pack ROM consists of one 2732 type 4K ROM which fits in the spare ROM socket of the Atom pcb. It gives the user a varied set of editing commands which enable you to create and edit text files while freeing you from the inadequacies of the BASIC line editor.

First Impressions

The Word Pack ROM comes imprisoned in a neat cardboard and expanded polystyrene box which should protect it against anything the Post Office are likely to throw at it. On opening the box you are confronted with details of how to fit the Word Pack ROM into the spare socket on the PCB. The ROM itself is supplied in a piece of antistatic foam.

The manual is well produced and generally adequate, although perhaps it does drop the complete novice in at the deep end to some extent. However, a certain amount of playing about will soon give the user familiarity. Entering the editor is simplicity itself. You merely type the single word EDIT in response to the prompt, and you're there.

Does it Work?

While the Word Pack is designed to be used with almost any Atom, and does not require the presence of disks, it is perhaps inevitable that many of the featurs one may expect to find in a large system are not present, in particular facilities like inserting blocks of text from disk or tape. Most facilities are availiable though.

In order to use the Word Pack, you must have 6K RAM in the upper txt space and it is advisable to have a full 12K machine. The floating point ROM is not required. One particularly pleasant feature is the ability to insert printer commands in the text to control formatting of the text. The printer controls availiable include page numbering, page length, paper width, right justification (on and off) indentation, centreing of titles, and also a facility for sending control characters to the printer to control such things as underlining, double width characters etc.

It would appear to be possible to use most makes of printer with this ROM as most facilities are not printer dependent. However, when testing the ROM with a Centronics 737 printer, it became clear that three facilities on the printer could not be used.

Firstly, it is not possible to underline a centred title as it is found that the underlining includes the space between the left margin and the title! One can Snopake it out of course, but that isn't the point. The second limitation is that the 737 has a proportional spacing feature which also can't be used without messing up the right justification. Lastly, it is possible with the 737 to expand a single section of a line, underline it etc. However, as the Word Pack requires all printer controls to be at the beginning of a line this advantage is lost.

In fairness, it must be said that Acorn couldn't possibly cater for all the peculiarities of all printers, but nonetheless it is a shame.

Using the editor facilities is generally speaking a pleasure. Editing is carried out on screen by using the cursor control keys and then using the editing functions A and B (insert text after and before the cursor) E (ditto at the end of text) R (replace) T (transfer) X (exchange) I (insert single character). Upper and lowercase characters are displayed by using the high resolution graphics screen. Other functions include move cursor to start of text, to end of text, next page, previous page, find a string, find a string and replace it with another, find a string and delete it , and the full set of COS or DOS commands. The only criticism I really have of the editing facilities is that the automatic repeat on the cursor control keys does not seem to delay at all between the key being pressed and autorepeat starting, and it is also far too fast.

Conclusions

The Word Pack ROM is certainly easy to use when you are used to it, although it does take a little time to become really accustomed to it. While it does not have the facilities to compete with the more sophisticated wordprocessors it should be perfectly adequate for most routine applications - particularly if your typing accuracy is anything like mine.

Vital Statistics

To use the Word Pack, you must have 6K RAM in the upper text space and it is advisable to have a full 12K machine. The floating point ROM is not required. Price 29.90 inc p&p, VAT. Supplier Acornsoft Ltd, 4a Market Hill, Cambridge. Delivery they say is ex stock.

MANCHESTER REPORT

Clem Rutter

There are few problems in setting up a user group and the results are most satisfying. I took the initiative, on the grounds that someone had to and circularised all the names I found on the national group's address list for Manchester. I proposed a meeting at the FE college where I work, offering to lay on power points and a couple of TV sets. Perhaps 10 people turned up to the first meeting in January and everyone wanted another, so we now meet on the last Tuesday of each month. After paying a 10 pound introduction fee, we get the room free for being a valid community interest group.

Our meetings are informal, everyone does their own thing. Some folks come alone, some bring fathers, some sons. Essentialy it is an ideas workshop. Interesting programs are swapped, and there is a thriving debugging group. Lately quite a bit of time has been spent working out what programs actually do. When pirating a program, or even obtaining it legally, get a copy of the documentation! Some of our members are doing some development work, and having a machine code expert at hand is a great boon.

One activity is standard at all meetings - discussing the supply problem. We are not amused by five month delays - someone is promising to bring champagne when his colour board arrives. I think that a useful function a local committee can perform is to locate local suppliers who can ferret out leads on where to get bits and pieces.

RGM writes -

Since this piece was written, Clem has left Manchester for the Medway Towns (Rochester, Chatham etc). The Manchester group is now being organised by John Ashurst. Anyone interested should contact him on 061-370 4962 eves.

Anyone who has started a local group or intends to do so, may we offer you the best of luck and please let us know so we can give you a plug in the Newsletter!

Finally, for users in Sweden, a user group has been formed. For more info, contact JANNE SODERBERG, Frihetsvägen 32, S-175 33 JÄRFÄLLA, Sweden.

Noise Free Graphics

In AT&P, the use of the WAIT command is encouraged to give noise free screen plotting. Unfortunately, this depends on the fact that the following PLOT, DRAW or MOVE command executes during the frame flyback period and thus noise is not displayed. When the DRAW is a complicated one (for instance, you might have a statement like DRAW (%(100*COS%T)),(%(100*SIN%T))) the execution period is so long the WAIT doesn't work.

The cure is to force a WAIT before actually doing the plotting but after calculating the values of the expressions. This can be achieved by, before doing any plotting, performing either manually or within the program the following.

Step 1: !£80=£022D80A9;!£84=£4CF9D0B0

Step 2: Poke the following values into £88 (ie !£88= one of the following:
mode 0 £F6E2 mode 1 £F73B mode 2 £F754 mode 3 £F76D and mode04 £F7AA)
Step 3: AFTER the CLEAR n statement in your program do !£3FE=£0080

This works by replacing the address of the point plotting routine held in £3FE with the following code:

```
LDA @£80  Wait for flyback signal
AND £B002
BNE £80
JMP pppp  address of point plotting routine
```

WA Chadwick, Bristol

New Products

I have just recieved a new catalogue from Acornsoft. The are now doing a range of non - games software as well as an extensive range of games including some new ones such as Space Invaders. Other software includes Soft VDU (an improved version of HWRITE), a utility pack comprising a disassembler, 1200 baud cassette interface, and a renumber, the FORTH language, a database system, some business-oriented programs, some maths utilities and a computer teaching aid (PEEKO).

Prices are generally £10 per cassette except as follows:
ATOM Business Book £6.95, tape £7.50
FORTH Book £5, tape £10
WORD PACK (in EPROM, not cassette) £26.00
All prices plus VAT, p&p. For more details, contact Acornsoft, 4a Market Hill, Cambridge CB2 3NJ.

DP Saville, of 16 Zulla Rd, Mapperley Park, Notts is offering a range of games for the Atom along with some utilities and a text processor. Price £3.00 per program plus p&p (£20 for the wordprocessor). He is offering a 10% discount to User Group members.

Lastly, a Toolkit ROM for Atom (reviewed elsewhere in this issue) is availiable at £35.00 fro WILLOW SOFTWARE, PO Box 6, Crediton, Devon EX17 1DL.

Index to Previous Issues

Newletter 1:TV interface problems. Contributing material. Operator precedence within Atom Basic. Useful subroutine. (printing string within assembly language). Efficient use of memory. Backus Naur Form. Premature exiting from FOR NEXT and DO UNTIL loops. Blunders.

Newsletter 2:Atom power supply. BASIC program storage format. Keyboard Input Routines. Back Issues. New Products. More useful addresses. Serial printer interface. Local user groups. Membership lists. More errors in AT&P.

Newsletter 3: Cassette Recorders. Loading hex dumps. More on wobbly telly. Bugs in Atom BASIC. More useful addresses. More on Keyboard input. Local groups.

Prices

Prices of software etc from the User Group library are as follows:

Cassettes £1.00 per cassette (1 program only per tape) Includes listing
Listings £0.15 per program
Address lists £0.15 per county
Hardware etc £0.15 per copy - see below

Hardware designs etc availiable include

TV to video monitor conversion
On Board Memory expansion beyond 12K
Interfacing ATOM to Commodore Printer
Cursor Control with individual keys
Using the Atom with an Epson TX80B Printer
13K Static RAM card for Atom
Atom Time Trials - the BASIC timed
An Analogue Voltage Interface and Joystick Controller

The Back Page

This section is about the size it should be this month...

AT&P p162 FDIM.
The example should read FDIM %JJ(5) and not DIM %JJ(5).

NL3 p4. Hex Dumps.
Modify the program as follows:

```
Insert 45 L=LENB
Change 50 IF L<>0 GOS.a
       130 IF L=1 Z=1
Delete line 141.
```

Software Library

Many thanks to all of you who have sent in contributions. Rather than let the library grow too large I am trying to keep it at about the same size while increasing the quality and diversifying the range.

Please note that we have decided from now on only to supply certain of the longer programs on cassette. This is because we have found that the logistics of recording large quantities of cassettes are impossible and this should allow at least a faster turnround.

I hope that you don't think we change the format of the library around too often but as the group grows it gets harder and harder to mantain an acceptable standard of service.

Programs availiable on cassette are marked with an asterisk (*).

New Titles

3D OXO

Play 3D Os and Xs against the Atom. On a 4x4x4 matrix. Tricky except for those hypermaths who play it on a 4x4x4x4 matrix. 4K.

EDITOR

Allows pages of text to be created and edited ready for outputting to the printer. 4K.

LIFE

Very fast version of Conway's game, simulating Life, the Universe, and Everything. Doesn't take seven and a half million years to run either. 4K.

CONTACTS

Allows names, addresses and 'phone numbers to be stored away on tape for subsequent retrieval. Ideal for those like me who write phone numbers on the backs of envelopes and then throw the envelope away. You can now record Top of the Flops on it instead. 12K.

DUNGEONS *

Similar to the popular game Adventure. Involves navigating round an underground dungeons while being attacked by ghouls, ghosts and ghastlies. 6K.

FINANCIAL

Calculats mortgage rpayments, interest on short term loans, etc . Causes sleepless nights. 3K.

Please note all sizes are approximate!

UTILITIES

RENUMBER

A fast renumber routine written is assembly language suitable for all sizes of Atom.

PACK

A machine language utility for reducing the size of a programme by condensing multiple spaces into a single space. This method ensures that syntax errors due to missing sifnificant space cannot occur, and readability is improved although the space saving is less than that obtained by eliminating all spaces.

Please note that although both programmes will run on an unexpanded Atom (both run in screen memory) the programme will need assembling in stages to run in a 2K Atom, as the source is of over 512 bytes. No problems should be encountered in Atoms of over 2K.

PONTOON

An Atom version of this popular game.

TEXT 3K
VDU 1K

SHARP

One for those with elephantine memories. You have to remember some numbers that have been displayed for a short time. But as you get better it gets harder.

VDU 512 BYTES
TEXT 1K

SPEED BOAT

Navigate your way round a lake avoiding various obstacles using the appropriate nautical terms.

TEXT 4K

OTHERS

ETCHA

Allows you to "draw" pictures on the screen (like the Etcha-Sketch toy). Works in graphics Ø.

TEXT 512 bytes
VDU 512 bytes

SPIROGRA

Draws Spirograph patterns in graphics mode 4. Requires full 6K screen memory, 1K lower text space and floating point ROM.

DEFFEQ

Solves second order linear differential equations and plots their response to a step input in graphics mode 4. Requirements as SPIROGRA.

SERIAL

I/F Serial Teletype/VDU driver to replace the inbuilt memory mapped display of the Atom. Written in assembler, it is suitable for all Atoms of over 4K which have at least 4K available in a continuous block. (for the source code). The binary code is less than 256 bytes.

Any Atom with the lower text space in operation can run the binary. Either the display only or the keyboard and display can be replaced. Requires the 6522 VIA and a modicum of external circuitry to drive either a RS232 or loop terminal.

SWUART

The software UART routine used to provide serial i/o in the above. As above, but no control code recognition or operating system interface. Hardware requirements 3K RAM, VIA, minimal external circuitry.

CONFACTORS

A programme for giving approximation of required ratios for gearing in machine tools and pairs of factors to help in selecting suitable pick-off gears. Not available on cassette.

ATOM USER GROUP

Newsletter 5 February 1982

Welcome once again to the Atom User Group Newsletter. The Group has expanded at a steady rate and now stands at approximately 2,500 members. As most people are probably aware Richard Meredith has gone into business and will no longer be involved in running the Group. I would therefore like to take this opportunity to thank him on behalf of all users for the effort he has put in over the past year in establishing the Group. As a result of this change, I will now be producing the newsletter and in order to maintain the same standard I would appreciate any contributions from members who have developed items of hardware or software and are prepared to write articles for inclusion in the newsletter. Many thanks to all of you who have written in with interesting ideas, they are all appreciated.

Unfortunately Newsletter 4 was distributed later than expected and in many cases people were not aware that we had a stand at the PCW show at Hammersmith in London during September. Thanks to all those who attended for making the three days a most enjoyable experience.

It is quite clear that many Atoms are being sold abroad judging by the requests for membership from countries all over the world. A special welcome is extended to all these users. At home it is nice to see the formation of a number of local groups. The addresses of two recently formed ones are enclosed in this newsletter.

Apologies to all those who have waited some time before receiving requests for software. As I think most people are aware the Group is run on a voluntary basis and on certain occasions, due to the volume of correspondence, turnround can be particularly slow. I can assure you however that steps are being taken to improve the situation.

I would be grateful if any general enquiries concerning technical points or software on order could be forwarded by letter instead of telephoning me at my home address in the evenings.

Finally, due to an oversight in the last newsletter we forgot be put in our address! For the benefit of all those that don't already know, it is:

18 Frankwell Drive,
Potters Green,
Coventry, CV2 2FB.

PETER FROST

CONTENTS

How to Write Special Printer Routines for the Acorn Atom

Introduction

Whenever you write a character on the screen and/or to a printer either in direct mode or in a program, the Atom executes a JMP (WRCVEC) i.e. it looks at memory locations # 208 and # 209 to find the start address of the routine which this character should be sent. This is illustrated in Figure 1. If you wish therefore to add your own routine you need first to change the WRCVEC pointer to point to the start of your routine, and then at the end of your routine jump back into the Atom's own routine.

N.B. Normally this routine resides at # FE52, but if the Wordpack is being used it is # ACCE, therefore the safest thin to do is to remove the old WRCVEC and store it elsewhere and then at the end of your routine use this as a pointer to the Atom's routine, as illustrated in Figure 2.

What has to go in your special routine?

All line numbers quoted below refer to the sample routine given in Figure 4 which is for a QUME SPRINT 5.

1. Initialisation (lines 40-70)

Remove the old WRCVEC and put in the pointer to your routine, store the old WRCVEC for use at the end of your routine, set up the VIA appropriately as output.

2. Restore (line 280)

This is not absolutely necessary as pressing BREAK restores the values of all the vectors.

3. Main Body of the Routine

This consists basically of saving the 6502 registers (line 180), dealing with the individual character (lines 210-270) and then retoring the registers (line 200) before jumping back into the Atom's routine. The way the character is dealt with is shown as a flow diagram in Figure 3.

Notes

1. If a control B or C is detected, it must not be sent to the Atom routine, or the Atom would wait for the appropriate response from a Centronics printer which would not be forthcoming. Therefore the control code is replaced by either a null (Ø), or an ESC (27).

2. Some printers have no auto linefeed, therefore it is advisable before sending out the character to compare it with the "character not sent to the printer" which is stored by the Atom at #FE, and which is initiallised on BREAK to #ØA - a linefeed character (line 260).

3. The routine shown in Figure 4 allows the Atom to drive a Qume Sprint 5 via PAO as an output with the Qume's "data terminal ready" line applied to PB7. It is set to work at 1200 baud and needs a MC1488 or equivalent running from, say two 9V batteries, to provide the RS232 standard drive - for more details see SWUART in the library.

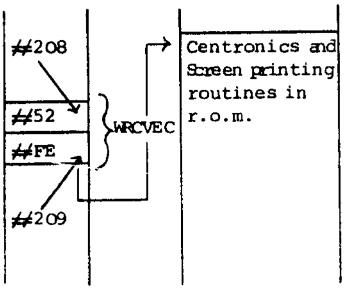
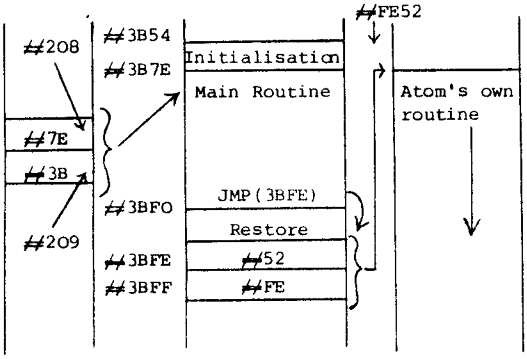


Figure 1



Paul E. Beverley

```
 10 I=#3B54;M=#3BFC;B=#B800;V=#208
 20 DIMLL9;FORN=0TO9;LLN=#FFFF;N.
 30 P.$21;FORN=1TO2;P=I;[
 40:LL0 LDAV;STAM+2;LDAV+1;STAM+3
 50      LDA@(LL5%256);STAV;LDA@(LL5/256%256)
 60      LDX@1;STXB+3;DEX;STXM+1              STAV+1
 70      LDA@13;STAM;JSRLL1;RTS
 80:LL1 LDX@0;STXB+1
 90      LDAM;ORA@#80;LDX@10
100:LL2 JSRLL3
110      SEC;RORA;ROLB+1
120      DEX;BNELL2
130      RTS
140:LL3 PHA;TXA;LDY@10
150:LL4LDX@14;DEX;BNEP-1
160      DEY;BNELL4
170      TAX;PLA;RTS
180:LL5 PHP;STAM;TXA;PHA;TYA;PHA
190      BITB;BPLP-3;JSRLL6
200      PLA;TAY;PLA;TAX;LDAM;PLP;JMP(M+2)
210:LL6 LDAM;CMP@2;BNELL7
220      LDA@128;STAM+1;LDA@27;STAM;RTS
230:LL7 LDAM;CMP@3;BNELL8
240      LDA@27;STAM;LDA@0;STAM+1;RTS
250:LL8 BITM+1;BMIP+3;RTS
260      CMP#FE;BEQP+5
270      JSRLL1;RTS
280:LL9 LDA@#52;STAV;LDA@#FE;STAV+1;RTS
290];N.;LINK I
300 P.$6$2"INITIALISE"&I'"BYTE O/P"&LL1'
310 P."RESTORE"&LL9'
320 @=1;P.$3" *SAVE QUME "&I" "&P+1" "&I'
330 P.$3;E.
```

Atomic Pulse Generator

Provided you have a 6522 VIA, and can solder a link onto you p.c.b., you can use the Atom to generate a T.T.L. pulse whose width and duration are programmable. The link that needs to be made is link 2 which joins the IRQ output of the VIA to the Atom's IRQ line.

How it works

The 6522 has two timers both of which are capable of generating interrupts and one of which can produce an output on PB7. This goes low when timer 1 is loaded, and goes back to logic 1 when the count reaches zero. Since the decrementing of this timer takes place at the clock rate (1 MHz), the idea is that the number loaded into this timer sets the time for which the output is low, and that the overall time of the pulse repetition is set by loading a (larger) number into timer 2. When this timer reaches zero an interrupt is generated, the response to which is to load the original numbers back into the two timers so that the whole process starts again. Since this is working on interrupts, the processor can be continuing to attend to various other tasks while the pulses continue to be produced.

The Program

```
10   P=#3BF0;I=#204;B=#B800;M=#80
20   ?I=P%256;I?1=P/256
30   B?14=160;B?11=128
40  (LDA M;STA B+5
50   LDA M+1;STA B+9
60   PLA;RTI;)
70   ?M=8;B?4=0
80   M?1=16;B?8=0
90   B?9=16
100  END
```

P = Start address of interrupt routine
I = Address of interrupt vector, IRQVEC
B = Base address of the VIA
M = Somewhere in zero page to store the high bytes of the numbers to be put into the timers. (Low bytes can be loaded into the appropriate low order latches - see lines 70,80.)

Notes

1. To stop the pulses going out you can disable PB7 by using B?11=0, but the interrupts continue to be serviced. To stop them altogether use B?14 = 32 - this makes the processor ignore the interrupt from timer 2 until B?14 = 160 is executed when the whole process will start again.

2. The actual width of the negative pulse is N + 1½ microseconds, where N is the number stored in timer 1.

3. The overall length of the pulse is N + at least 30 microseconds where N is the number loaded into timer 2. The extra time is that which is taken is servicing the interrupt, and also there are times when the processor decides it does not want to be interrupted and ignores the IRQ until it is ready to service it. This means that there will be some slight instability in the output pulse.

4. A very powerful application of this idea is to average the output pulse using for example a four pole Butterworth filter (!) and use the pulse width to generate a programmable d.c. level i.e. a digital to analogue convertor. With an appropriate filter costing about £2, and using a pulse width of 4096 you have a 12 bit D to A using only one output line instead of 12 and costing a fraction of the £30 for a proper convertor. The only drawback is that it takes time to average the pulse out to a steady enough d.c. level, therefore the response time is about 20 milliseconds.

For more details of the filter circuit, or for a copy of a seven page explanation of the operation of the 6522 counter timers including ideas for electronic organ and frequency meter programs, send a S.A.E. to: Paul E. Beverley, Norwich City College, Ipswich Road, Norwich, NR2 2LJ.

DATA TRANSFER ON THE ATOM

A number of ATOM users appear to have trouble with matching their cassette recorders to the input and output levels on the ATOM without corruption of data. After many hours (days) I found the following methodof set-up to be fairly easy, reliable and allows for the change of cassette recorders.

The instruction on page 8 of ATP provides a quick method of obtaining a rough level for the input/output of the ATOM, however it is not routed through the COS so it does not really help to detect corrupt data transfer. ATOM were fairly helpful and supplied some further tips and the following more stringent test.

The test is done by saving the contents of the 4k byte of the basic ROM which gives a test tape lasting approx. 4 minutes.

```
*SAVE "TEST TAPE" COOO DOOO
```

On rewinding the tape a catalogue of the tape should show 16 consecutive blocks as shown below.

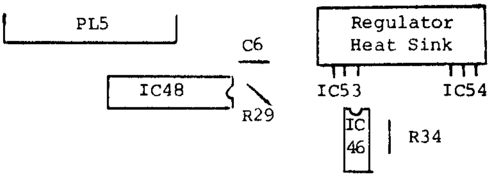
```
            *CAT "TEST TAPE"

(from)  TEST TAPE COOO COOO OOOO FF

( to )  TEST TAPE CFOO COOO OOOF FF
```

If all is well the tape should be read by the ATOM and if no errors found the angular cursor should reappear after approx. 4 minutes.

However if your cassette recorder is like mine it will cause a number of errors. I therefore spent some time changing the values of R29 and R34, but to no avail. However the following modifications to the ATOM has produced completely error free transfer of data.

R29 was removed and replaced by a 1 meg ohm vertical sub-miniature carbon preset.This will adjust the output from the ATOM to the cassette recorder.

R34 was removed and replaced by a 220k ohm vertical sub-miniature carbon preset. This will adjust the input to the ATOM from the cassette recorder. One leg of the presets are cut off and short lengths of wire are attached.



The test on page 8 of the ATOM manual, then recording the tape test adjust R29 until the recording level meter on the cassette shows 70% (-3dB). On playing the tape adjust R34 until a continuous stream of "X" are printed out. You can then carry out the above test to ensure the COS is operating correctly.

If you use a cassette recorder with an automatic recording level you cannot adjust R29 very easily as you probably do not have a recording level indicator. The only way round this is to record the test on page 8, adjusting R29 in steps of ⅛ turn every 15 secs., then play the tape on a normal Hi-fi cassette recorder, check which position shows 70% level on the meter(s). This can be done by counting the change in level i.e. the 3rd level will be approx. 1/4 turn on the pot from the max end. R29 can then be adjusted to the appropriate setting, (max ⅛, 1/4 turn). The tape can then be played in the normal way and R34 adjusted until a continuous stream of "X" printed out. Then the long test tape can again be run. On some small test recorders with a tone control, which effects the low level output it may be necessary to adjust it, with a bias towards the treble setting. I have found this necessary in particular when loading pre-recorded ACORNSOFT tapes.

I have set up some of my friends machines using this method with 100% success.

C. Garrett

TIMER FOR ATOM

This article gives brief notes on how to derive a standard time interval for use in producing clock or stop watch facility.

Hardware requirements - VIA chip   Memory requirements - minimal
The unit provides a 1Hz or 60Hz signal (software selectable) which is used to decrement counter 2 of the VIA chip. The unit times independantly of the CPU.

Using the Unit

To use the unit bit 5 of the Auxiliary Control Register must be SET. This is done by ?#B80B = #20. The timer then counts pulses on P86. If CB2 is SET the unit will work at 60Hz, if it is RESET it will work at 1Hz CB2 is set or RESET as follows. SET - ?#B80C = #E0 RESET ?#B80C = #C0

1. Timing a Section of the Program

At start of program set counter to #FFFF by ?#B808 = #FF; ?#B809 = #FF

T = !#B808&#FFFF
. . . .

section of program to be timed
. . . .

T = T-!#B808&#FFFF

T will be equal to the time taken for the section of program either in seconds or jiffies (60 jiffies = 1 second) depending on the setting of CB2.

2. Providing a Time Delay

Set timer to time required
DO UNTIL ?#B80D&#20 = #20

B80D is the interrupt register and bit 5 is set when timer reaches zero ONLY IF the timer has NOT been read, i.e. by A=?#B808 etc.

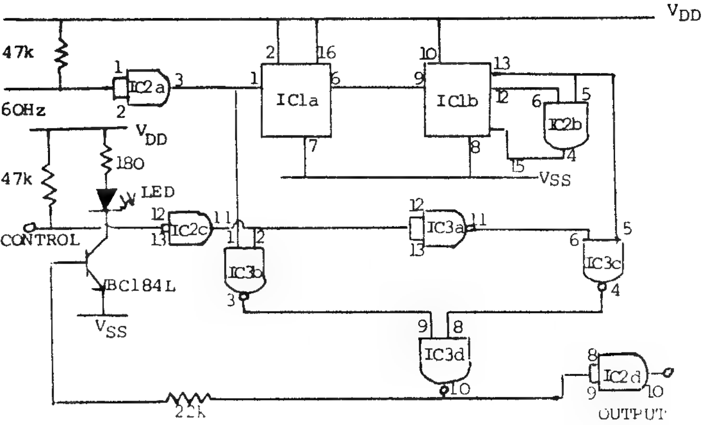
3. Printout of Clock on VDU

?#E1 = 0; REM remove cursor
PRINT $30, !#B808&#FFFF
main loop of program
to provide an UP counter replace PRINT $30, !#B808&#FFFF by PRINT $30, !#B808&#FFFF:#FFFF

Provides a display of clock counting DOWN in top left corner of VDU.

Modification

Only one modification is required to ATOM - run a wire from pin7 of the video output socket to pin A1 of the 64 pin connector.



IC1 divides the 60Hz video signal by 60 to give a 1Hz output. IC2 is a buffer for both inputs & outputs. IC3 selects the output (1 or 60Hz). D1 will be on for a 60Hz output or will flash at 1Hz for a 1Hz output.

Connections		Note
60Hz in	pin A1	IC1 = 4518B
Control	pin A11	IC2 = 4081B
Output	pin A4	IC3 = 4011B
V_{DD}	pin B1	V_{DD} = +5V
V_{SS}	pin B32	V_{SS} = 0V

A & B are shown reversed on ATOM circuit diagram.

Current consumption 10 mA with LED.

IPB

REVIEW - PROGRAM POWERS TOOLBOX

As a follow up to the review of the Willow Software Utility ROM in NL/4 we now present a review of Program Powers Toolbox. A comprehensive set of 29 commands are available. The Toolbox comes in a 4K EPROM which fits into the socket 24 on the Atom PCB. If you already have a chip installed in this position at the end of this article we give the name of the supplier for mounting a number of such utilities. Installation is very easy and initialisation of the Toolbox is initiated by LINK #AFØØ. All the commands listed below are now available. Probably one of the most useful facilities is a 1,200 baud cassette interface. This facility works very well although increasing the baud rate will reveal any weak links you may have in you Atom cassette interface. The only advice I can offer is to use good cassettes and keep your head clean! At the 1,200 and 300 baud rates each byte received or transmitted is displayed at the top right hand corner of the screen.

A brief summary of all the commands available:

FIND "string" List the line numbers which contain "string".

RENUMBER X,Y Renumbers all lines starting from X in steps of Y including GOTO's and GOSUB's and list the line numbers containing indirect jumps which it cannot renumber allowing you to edit the appropriate line.

AUTO X,Y Automatic generation of line numbers starting at X in units of Y.

DELETE X,Y Delete lines from X to Y.

VAR Prints values of all variables A-Z (hex option available) on the screen.

LVAR " " " " " " " " " on a printer.

ZERO Sets all variables A-Z to zero.

HEX YYYY Prints blocks of 32 bytes from YYYY in HEX and in ASCII. Space prints next block.

INHEX YYYY Tabulates HEX codes in memory in instruction format i.e. 1,2 or 3 bytes appear as data depending on the instruction represented by the 1st byte.

STEP (VIA required for this command and the IRQ link must be in place.) This allows single stepping of a program on a line by line basis with the current line being displayed at the top of the screen, The program halts at each line and waits for a key to be depressed before continuing.

TRACE X VIA and IRQ link needed, same as above but waits for a delay X before proceeding.

LTRACE Prints numbers of lines as executed on a printer.

OFF Turns off TRACE, LTRACE and STEP commands.

DUMP Dumps contents of the screen to a printer.

VECTOR X If X=O then baud rate = 300, if 1 then 1,200.

BEEP X,Y Generates a note of pitch X and duration Y, ideal for games.

CURSOR X,Y Positions the cursor at the Xth column on line Y.

KEY A The keyboard in scanned once and the value of the ASCII code of any key depressed is returned.

INKEY $A Similar to KEY but returns the character corresponding to the key pressed in the designated string.

STOP Inserted at a point in a program will halt execution.

POP Allows jumps from sub-routines to any point in the main program, not just back to the point after the GOSUB.

XIF..THEN..ELSE Very powerful command allowing statements like:

```
10 XIF A=O THEN PRINT "YES"
     ELSE PRINT "NO"
```

WHILE..ENDWHILE Statements between these two commands are executed until the condition specified in the WHILE command is true.

READ..DATA..RESTORE Makes up for the lack of these commands in Atom basic. These command allow strings of data to be defined in a user program.

ON ERROR Every time an error occurs the normal error message is suppressed and execution will recommence after the ON ERROR command.

As you can see the commands provided fill in any gaps in the existing Atom firmware and as such is highly recommended as a useful addition to your Atom.

NEWS....NEWS....NEWS....NEWS....NEWS

Acorn Computer Group North - Meets fortnightly at: Abraham Moss Centre, Cresent Rd., Salford.
Organiser: John Ashurst - Tel. 061-681-4962.

Sheffield User Group - Those interested in knowing more should contact Mr. A. Tinker
40 Middle Hay View, Gleadless Valley, Sheffield.

Swedish User Group - Details from Janne Söderberg, Frihetsvägen 32, S-175 33 Järfälla,
Sweden. Tel. 0758-31753.

BBC MICRO Local Group - For those in the Norwich area. Organiser: Paul Beverley,
Norwich City College, Ipswich Road, Norwich.

Any Radio Hams or SWL's interested in swapping information etc. should contact John Riggs who is setting up a Radio Amateur Atom User Sub-Group at: 68 Gomer Lane, Gosport, Hampshire.

Barry Husbands of lo Plas Tirion, Nant Parc, Johnstown, Clywdd, is trying to link an 8" disc-drive to an Atom, if anybody has succeeded in achieving this he would be interested to hear from you.

I recently received a letter from the Programs Editor of PCW who is offering good rates for any user who has written an interesting program that would be suitable for publication in this magazine. A paid referee register also exists for those competent to vet such software. Anyone who is interested should contact: Maggie Burton (Programs Editor), 14 Rathbone Place, London. W1P 1DE

Some interesting new products from Acornsoft include:

ATOM FORTH - A complete implementation of the Forth language. The cassette contains the Forth dictionary and compiler, the tape interface/screen editor, graphics package and high res. graphics demo.

WORD TUTOR - Consists of 3 versatile programs designed to aid the development of language abilities in children of primary school age.

ATOM CHESS - Provides 6 levels of play, all the usual facilities for replaying games and restarting from any point.

ATOM SYNTHESISER - Turns the Atom into a programmable synthesiser. You can record and play back 4 seperate tunes.

Further information can be obtained from: Acornsoft Ltd., 4a Market Hill, Cambridge. CB2 3NJ

Two useful items of software are available from Mr. D. Dupe, 6 Hindhead Rd., Reading, Berks. These are ATOM DISSASSEMBLER and ATOM TRACE, providing all the usual facilities.

HARDWARE & SOFTWARE LIBRARIES

Unfortunately we can no longer supply software on cassette. This is due to the difficulty in producing the cassettes and the logistics involved in distributing them quickly.

Prices of software etc. from the User Group Library are as follows:-

Listings £0.20 per program for U.K. users (£0.25 for overseas users).
Address lists £0.20 per county.
Hardware listings £0.20.
Back issues of Newsletters (1,2,3 & 4) available at £0.60.

The above items include P. & P. Please forward orders to the address on the Front Page.

Hardware Library

TV to video monitor conversion.
On Board Memory expansion beyond 12K.
Interfacing ATOM to Commodore Printer.
Cursor Control with individual keys.
Using the ATOM with an EPSOM TX80B Printer.
13K Static RAM card for ATOM.
ATOM Time Trials - all BASIC commands timed.
An Analogue Voltage Interface and Joystick Controller.

SOFTWARE LIBRARY

BIO — Find out just when you are at your intellectual physical and emotional peaks. TEXT 4k VDU 6k.

BOMBS AWAY — Like AIR RAID but uses high resolution graphics to generate the profile of a city. TEXT 4k VDY 6k.

BREAKOUT — Demolish a wall of bricks with a bat and ball, with sound effects. TEXT 3½k.

CONFACTORS — A program for giving approximation of required ratios for gearing in machine tools and pairs of factors to help in selecting suitable pick-off gears.

CONTACTS — Allows names, addresses and 'phone numbers to be stored away on tape for subsequent retrieval. Ideal for those like me who wrtite 'phone numbers on the backs of envelopes and then throw the envelope away. 12k.

DEBUG — A debugger for machine code programmes similar to the monitor found on machines like the Acorn System One and the Nascom-1. Written in a hybrid of Basic and assembler. Size 3105 bytes source, 30 bytes machine code. No additional hardware needed.

DEFFEQ — Solves second order differential equations and plots their responses to a step input in graphics mode 4. Requirements as SPIROGRA.

DODGEMS — Navigate your way through a maze collecting points on the way while being chased.

DOMINOES — Play against the Atom with this version of 5's and 3's. TEXT 4½k VDU 6k.

DUCKSHOOT — You have just disturbed a field of ducks and have to shoot them before they fly out of range.

DUNGEONS — Similar to the popular game Adventure. Involves navigating round an underground dungeon while being attacked by gouls, ghosts and gastlies. 6k.

EDITOR — Allows pages of text to be created and edited ready for outputting to the printer. 4k.

ETCHA — Allows you to "draw" pictures on the screen (like the Etcha-Sketch toy). Works in graphics Ø. TEXT 512 bytes VDU 512 bytes.

FINANCIAL — Calculate mortgage repayments, interest on short term loans, etc. Causes sleepless nights. 3k.

FIND SNOOPY — Find Snoopy lost in a forest using only compass points to direct you. TEXT 2k.

HWRITE — Allows mixing a high resolution graphics (mode 4) and text, by replacing the existing VDU routines with another which creates the dot pattern needed to form 64 character ASCII subset, as well as recognising most of the Atom's control codes. 12k.

LIFE — Very fast version of Conway's game, simulating Life, the Universe, and Everything. Doesn't take seven and a half million years to run either. 4k.

LUNAR LANDER — Land your space craft on a randomly generated landscape which enlarges as you get closer. TEXT 3k VDU 6k.

MAZE — Navigate your way out of a maze using 3D projection of its passages to help you. TEXT 4k VDU 6k.

MISSILE — Use skill and judgement to eliminate passing ICBM's. TEXT 1k VDU 512 bytes.

MOONLANDER — Given a certian amount of fuel you have to land your spacecraft gently on the moon. TEXT 1½k.

MOUSE TRAP — Catch a mouse by "drawing" a trap round it as it moves over the screen. TEXT 1.5k VDU MODE 1.

OTHELLO — Play against the Atom or a friend using the fast response time game of othello. TEXT 4½k VDU Graphics 3.

3D OXO — Play 3D Os and Xs against the Atom. On a 4x4x4 matrix. Tricky except for those hypermaths who play it on a 4x4x4x4 matrix. 4k.

PACK A machine language utility for reducing the size of a program by condensing multiple spaces into a single space. This method ensures that syntax errors due to missing significant space cannot occur, and readability is improved although the space saving is less than that obtained by eliminating all spaces. Please not that both programmes will run on an unexpanded Atom (both run in screen memory) the program will need assembling in stages to run in a 2k Atom, as the source is of over 512 bytes. No problems should be encountered in Atoms of over 2k.

PASSE-TEMP Play against the Atom in this version of "Connect 4". TEXT 3k.

PONTOON An Atom version of this popular game. TEXT 3k VDU 1k.

RENUMBER A fast renumber routine written in assembly language suitable for all sizes of Atom.

SERIAL I/F Serial Teletype/VDU driver to replace the inbuilt memory mapped display of the Atom. Written in assembler, it is suitable for all the Atoms of over 4k which have at least 4k available on a continuous block (for the source code). The binary code is less than 256 bytes. Any Atom with the lower text space in operation can run the binary. Either the display only or the keyboard and display can be replaced. Requires the 6522 VIA and a modicum of external circuitry to drive either a RS232 or loop terminal.

SHARP One for those with elephantine memories. You have to remember some numbers that have been displayed for a short time. But as you get better it gets harder.
VDU 512 Bytes TEXT 1k.

SPEED BOAT Navigate your way round a lake avoiding various obstacles using the appropriate nautical terms. TEXT 4k.

SPIROGRA Draws Spirograph patterns into graphics mode 4. Requires full 6k screen memory, 1k lower text space and floating point ROM.

SWART The software UART routine used to provide serial i/o in the above (serial). Same as serial but no control code recognition or operating system interface. Hardware requirements 3k RAM, VIA, minimal external circuitry.

ATOM USER GROUP

Newsletter 6 July 1982

Welcome once again to the Atom User Group Newsletter. A number of members are concerned that as Newsletters do not come out every 3 months as we had hoped, they will not be entitled to receive 4 during 1 years membership. To allay any fears membership covers 4 Newsletters even if the period over which they are produced exceeds 1 year. I will of course endeavour to bring out a Newsletter as regularly as possible.

I have also had a number of queries concerning the renewal of membership. The simplest thing here is that once you have received 4 Newsletters and wish to renew, send a cheque for £4.00 plus your current (and old address if you have moved) to the address below.

Unfortunately due to the lack of help and also heavy work commitments I will be unable to run a stand at this years PCW show in London, apologies to all those hoping to see us there.

In the last Newsletter I made a special welcome to users abroad. I would now like to say a special hi! to all Canadian members since applications to join us have been coming in thick and fast.

Once again thanks to all of those who have contributed to this issue. If anyone out there has anything they think may interest our members please forward the relevant information to the address below.

I unfortunately managed to get into a terrible mess some time ago with certain orders for software and some people still have not received what they asked for. I am trying to refund all those who asked for cassettes just as we stopped selling them and to sort out any other problems with orders.

Please note I do not accept telephone calls at my home address, any queries must be forwarded by letter. For those of you who don't know my address, it is:-

18 Frankwell Drive,
Potters Green,
Coventry. CV2 2FB

PETER FROST

CONTENTS

ATOM ECONET - A SPECIAL ARTICLE

ECONET was one of the first networking systems to become available to microcomputer users, and is unique in that it is available on a very low cost computer such as the ATOM. While ECONET may be used on a network system consisting entirely of ATOMS, it will more usually be found on a system consisting of up to 254 ATOMS and a larger disc based system such as the system 3 & 4. The large system will be acting as a file server giving each ATOM the appearance of having a disk to itself. There may also be on the network a "printer server" which of course would allow a single printer to be accessed by all the ATOMS of the network.

The intriging thing about ECONET is that in general the ATOM running ECONET appear unchanged. The only (apparent) difference is that on BREAK the message "ACORN ATOM ECONET 1.48" appears instead of the usual one, and the "station number" appears in the top right hand corner. On executing a LOAD, SAVE or CAT however the ECONET wakes up and files are accessed from the file-server and printed output via the printer server.

Furthermore, we find that in the NOS (Network Operating System) 3 more commands are available. These are *VIEW, *REMOTE and *ROFF. VIEW copies the screen of a remote station to your own screen and REMOTE completely takes over control of the remote station so that you can input a program and run it on the distant machine without affecting your own, ROFF turns this function off.

This deals briefly with the NOS of ECONET. There is however a set of lower level commands - the network primitives - which are accessible via assembly language or BASIC. For a system consisting entirely of ATOMS it is these which are of most interest as there will presumably not be a file server on such a system, although there may be a printer server.

The network primitives are, as the name suggests, a set of primitive operations which can be built up into more sophisticated operations such as the NOS described above. These operations include TRANSMIT, RECEIVE, PEEK, POKE, START and RESET.

TRANSMIT and RECEIVE allow the station to transmit data into a station which is expecting it to receive data from another station which sends it. These operations require the co-operation of the distant station. In addition to receive the operator can specify which particular station he wishes to receive from on the net.

The other 4 operations are termed "intermediate operations" PEEK copies a block of memory from the distant station to the local station, and POKE performs the reverse. RESET places the distant station in a tight loop so that it will only respond to network operations. This allows the execution of complex sequences of network operations without any interference by any program the distant machine is executing. For indeed ECONET primitives will operate while the distant machine is running a quite separate program. Lastly START starts the processor in the distant machine, executing instructions at the address specified, this is the only way to get the distant machine out of the RESET state other than by pressing BREAK. None of these immediate operations require the co-operation of the distant machine.

As an example let's consider the *VIEW network command. This must consist of a PEEK to location ≠≠BØØ2 to find the graphics mode the distant machine is in (if the distant machine is in a graphics mode the local machine will be switched into the same graphics mode). This is followed by another PEEK to get the data in the distant machines memory (8ØØØ-81FFH for text or mode Ø, 8ØØØ-97FFH for mode 4).

HARDWARE - The ECONET hardware consists of one very long thin PCB (about 1½" wide by the length of the ATOM) which plugs into PL8 on the ATOM PCB. One EPROM plugs into the spare socket in the ATOM. All stations are connected together by a four core cable and the maximum distance between stations is about 1 mile. The four cores consist of two balanced feeders one carrying the network clock at around 200kHz and the other the data. The formatting of data is controlled by an MC6845 advanced data link controller. Don't try dissassembling the ROM to find out how it works, unless you are unusually persistent you won't.

See page 6 for conclusions.

USEFUL TIPS

Many thanks to all those who have contributed items of interest.

*RUNNING A PROGRAM WITH A DIM STATEMENT AND PREVENTING ERROR 30

Place the following before the DIM statements.

```
1Ø  ?≠≠ØD=Ø;    ?≠≠ØE=≠≠82;    ?≠≠23=Ø;    ?≠≠24=≠≠82
```

Terance Abrahams.

KEYBOARD DEBOUNCE - PREVENTS DOUBLE CHARACTERS

```
1Ø  P = ≠≠21C
2Ø  (;JSR≠≠FB8A;   JMP≠≠FE94;)       REM tenth of a second delay.
3Ø  ?≠≠2ØA=≠≠1C;   ?≠≠2ØB=≠≠2
4Ø  END                              ( ) =  Square brackets for assembler.
```

The patch routine works by jumping to a ROM routine which waits for a tenth of a second before using the normal get char. routine, and alters the BASIC pointers to include it.

EDWARD BLACK

INPUTTING EQUATIONS DIRECTLY INTO BASIC CODE

If it is required to input an equation for say plotting a graph, this can be done by inputting the equation into a string then transferring the numbers stored in the memory (ASCII code) into a line of spaces left in the program.

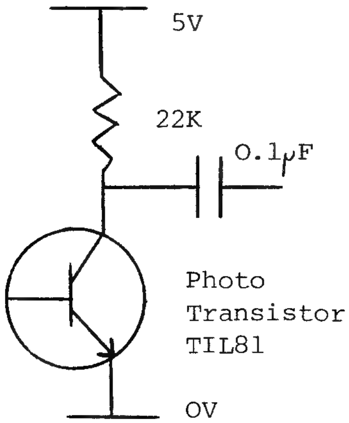
```
1Ø  DIM A9                                5Ø  (ELEVEN SPACES)
2Ø  INPUT $A                              6Ø  PRINT X
3Ø  X=5                                   7Ø  END
4Ø  FOR B=Ø TO 9;  B?≠≠2942=A?B;  NEXT B
```

Replying " X=X*2 " to the prompt will give 5Ø X=X+2 when relisted.

M. GERRARD

CHEAP AND SIMPLE LIGHT PEN FOR THE ATOM

For those of you with a VIA here is a cheap light pen for you to experiment with. You will have to write your own software to drive this but the following hints should help. (Note: Assembler routines must be used.) Start one of the timers in the VIA on receiving the flyback signal, set the IRQ vector (≠≠Ø2Ø4) to the address of a routine which stops and reads the timer. The count of the timer can then be used to find the approximate position of the pen. (Note: Link 2 will need to be completed.)



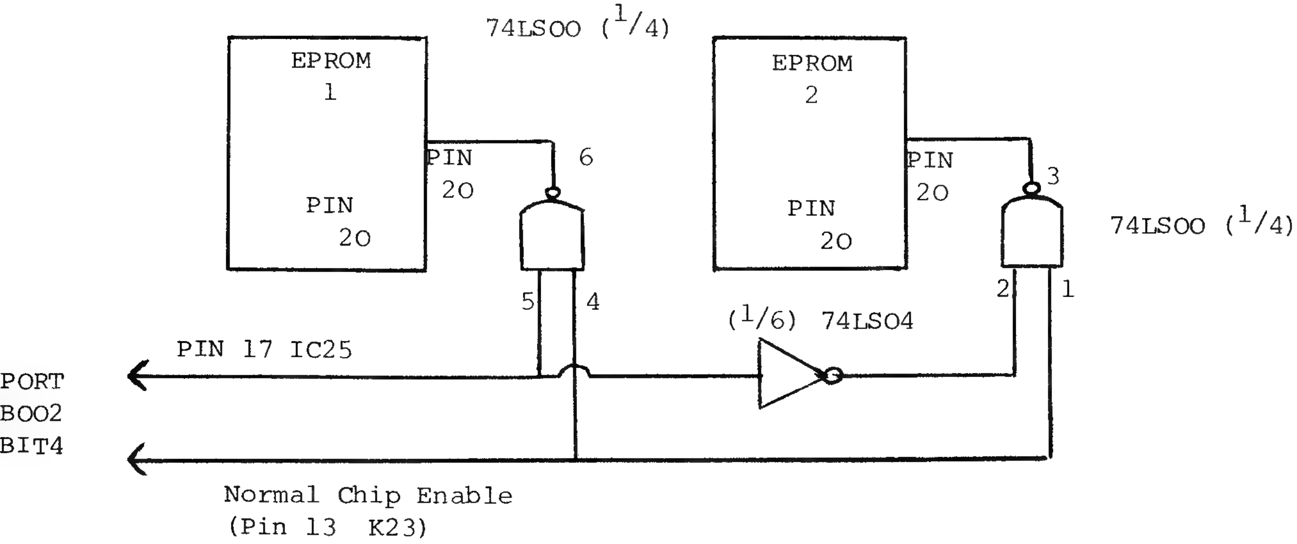
M. GERRARD

FAST ATOM

The 65Ø2A micro is a direct pin for pin replacement for the 65Ø2 in your ATOM but runs at twice the speed. The 2MHz clock can be obtained from IC44B (o/p A) and the original cassette tone appears at IC44 A o/p D.

M. GERRARD

UTILISING 2 ROMS IN ATOMS UTILITY SOCKET



With the great number of utility chips now available for the ATOM the problem is how do you install both in the ATOM at the same time. Here is a very cheap and simple way of achieving this:

The EPROM required is switched via software in the following manner:

BIT 4 = 0 gives EPROM 1 selected
BIT 4 = 1 gives EPROM 2 selected

K. ROBSON

GRAPHICS CHARACTERS ON A WHITE SCREEN

Use the following:

1Ø COLOUR Ø; CLEAR 4; P. $12 (it does the inversion better than the manual suggestion).

Also ?153 = X positions the cursor horizontally (X = 0 to 31)

and ?148 = Y " " " vertically (Y = 128 to 152)

D. ROWE

ROTATE - A USEFUL DESIGNERS AID

This is not just another 3D rotation program, it is a comprehensive visual aid to designers and technical drawing tutors. The program allows input from a taped file or by describing an object using a moveable cursor. Then a high resolution wire frame representation of a third (end) view is displayed followed by an isometric view and rotations around two axes simultaneously or on one axis Finally, an object may be recorded for use by the program later.

If, at the start of the run, you decide to enter a new object then press "D as instructed. Two cross-wires will appear on the screen. Around the top crosswire draw the side view of your shape. This is done by moving the flashing cursor using the keys: V=left, B=right, J=up, N=down. Only straight lines may be drawn, which is done by showing the computer where the two ends of the line are. To do this, position the cursor where the line is to go from and press "F" then move the cursor to where the line is to go to and press "T". When this side view is complete press "Q" and the cursor will jump to the lower cross-wire here the plan view is to be drawn.

The task is much easier now because the computer is only looking for the distance of each line up the screen and will find the X-distance itself. Now we come to the golden rule which must be obeyed implicitly while drawing. In each view every line (or edge) on the object must be drawn in the same order as on the first.

Once both views have been described, the end view is displayed by pressing any key; then if RETURN is pressed the isometric view will be shown. After thi pressing RETURN will rotate the object about the X and Y axes simultaneously. I rotation about the X axis only is required this mode is set up by pressing X and RETURN. If you are satisfied with the object you may at any point in either mode press "R" and RETURN and instructions for recording the object will be shown. This program occupies 2.8K of text and 6K of graphics and requires the floating point ROM.

Many thanks to Phillip Tubb for contributing such an interesting and novel program.

```
 5 DIM B(80),C(80),D(80),E(80),F(80),G(80),W(3),RR(2),P(-1)
 7 V=1;X=99;Y=150;L=1;@=100;J=2
10 GOS.470;P.$21
30[
31;RR0 JSR £FE71;STY£80;RTS;]
33 P.$6
35 CLEAR 4
36 MOVE128,136;DRAW128,152;MOVE128,40;DRAW128,56
37 MOVE120,144;DRAW 136,144
38 MOVE120,48;DRAW 136,48
40 REM keyboard  input
50sLINK RR0;A=?£80
60 IF A=42;Y=Y+1;G.a
70 IF A=34;X=X+1;G.a
75 IF A=46;Y=Y-1;G.a
80 IF A=54;X=X-1;G.a
81 IF J=1 OR A<>38;GOTO 88
82 P.$7
83 IF L=2;D?V=Y ;J=1;G.s
84 B?V=X ;C?V=Y ;J=1;G.s
88 IF J=2;G.91
90 IF A=52;J=2;G.b
91 IF @=V AND J=2;G.200
92 IF A<>49 OR J=1;G.100
93 P.$7
94 FOR X=1 TO 10;WAIT;WAIT;N.X
95 IF L=1;@=V
96 IF L=1;L=2;V=1;X=99;Y=48;G.s
100 PLOT13,X,Y;WAIT;WAIT;PLOT 15,X,Y;G.s
```

```
120b IF L=2;G?V=Y;A=D?V;G.130
123 E?V=X;F?V=Y;A=C?V
130 MOVE(B?V),A;DRAW (E?V),Y
131 WAIT;WAIT;WAIT;WAIT;WAIT
135 V=V+1;P.$7
140 G.s
150a PLOT 13,X,(Y+1)
151 PLOT 13,(X-1),(Y-1)
155 PLOT 13,(X+1),(Y-1)
160 WAIT;WAIT
170 PLOT 15,X,(Y+1)
171 PLOT 15,(X-1),(Y-1)
175 PLOT 15,(X+1),(Y-1);G.s
200 GOS.n
204 REM iii end elevation iii
205 CLEAR4
210 F.A=1 TO V-1
215 MOVE((D?A)+90),((C?A)-48)
220 DRAW((G?A)+90),((F?A)-48)
225 N.A
230 REM ii rotation routine ii
250 INPUT $W;%N=0.25;L=1
251 %N=%N+0.43
260 CLEAR4
270 FOR A=1 TO V-1
280 Z=(D?A)-48;X=(B?A)-128
281 Y=(C?A)-144;GOS.c
290 MOVE (128+K),(96+Q)
300 Z=(G?A)-48;Y=(F?A)-144
301 X=(E?A)-128;GOS.c
310 DRAW (128+K),(96+Q)
350 NEXT A
352 INPUT $W
354 IF $W="" ;G.251
357 IF $W="X";L=2;G.251
360 IF $W="R";GOTO e
361 END
370 REMiiiiii rotating iiiii
380c IF L=2;%Z=Z;K=X;G.y
382 %T=ATN(Z/X);%I=Z/SIN(%T)
385 %K=COS(%N+%T)*%I
387 %Z=SIN(%N+%T)*%I;K=%K
390y %T=ATN(Y/%Z);%I=Y/SIN(%T)
393 %L=COS(%N+%T)*%I;Q=-(%L);R.
470 P.$12
471 REM the start of the prog
475 P."         ORTHOGRAPHIC TO"'
477 P."         *3D* CONVERTER"''
480 PRINT"   choose mode"'
490 INPUT"'R'=RECORD;'P'=PLAYBACK"$W
500 IF $W="R";R.
505 REM tape file input
510 PRINT"   playback"'
512 FOR J=1 TO 40;WAIT;WAIT;N.J
515 PRINT"POSITION TAPE ON FILE LEADER"'
520 PRINT"play AND PRESS A KEY"'
525 LINK £FFE3
530 V=BGET X
535 PRINT"file found"'
540 FOR Y=1 TO V
550 B?Y=BGET X;C?Y=BGET X;D?Y=BGET X
```

```
555 E?Y=BGET X;F?Y=BGET X;G?Y=BGET X
560 NEXT Y
570 PRINT" loaded;PRESS A KEY"'
575 LINK £FFE3
577 CLEAR4
580 FOR Y=1 TO V
590 MOVE (B?Y),(C?Y);DRAW (E?Y),(F?Y)
595 MOVE (B?Y),(D?Y);DRAW (E?Y),(G?Y)
600 NEXT Y
602 V=V+1
605 INPUT $W;G.205
615 REM ii recording data ii
620e CLEAR 0
630 PRINT"  record object"'
632 F.J=1 TO 40;WAIT;WAIT;N.J
635 PRINT"PUT TAPE INTO record AND ALLLOW"'
640 PRINT"A LONG LEADER THEN PRESS A KEY"'
645 V=V-1
650 LINK £FFE3
660 BPUT X,V
665 WAIT;WAIT;WAIT;WAIT;WAIT
670 FOR Y=1 TO V
680 WAIT;WAIT;WAIT;BPUT X,(B?Y)
685 WAIT;BPUT X,(C?Y);WAIT;BPUT X,(D?Y)
690 WAIT;BPUT X,(E?Y);WAIT;BPUT X,(F?Y)
695 WAIT;BPUT X,(G?Y)
700 NEXT Y
710 PRINT" file recorded"'
720 PRINT"   PROGRAM FINISHED";END
730 REM iiii bug killer iiii
735n F.A=1 TO V-1
740 IF D?A=48;D?A=49
745 IF B?A=128;B?A=129
750 IF G?A=48;G?A=49
755 IF E?A=128;E?A=129
760 IF C?A=144;C?A=145
765 IF F?A=144;F?A=145
770 N.A;R.
```

```
120b IF L=2;G?V=Y;A=D?V;G.130
123 E?V=X;F?V=Y;A=C?V
130 MOVE(B?V),A;DRAW (E?V),Y
131 WAIT;WAIT;WAIT;WAIT;WAIT
135 V=V+1;P.$7
140 G.s
150a PLOT 13,X,(Y+1)
151 PLOT 13,(X-1),(Y-1)
155 PLOT 13,(X+1),(Y-1)
160 WAIT;WAIT
170 PLOT 15,X,(Y+1)
171 PLOT 15,(X-1),(Y-1)
175 PLOT 15,(X+1),(Y-1);G.s
200 GOS.n
204 REM iii end elevation iii
205 CLEAR4
210 F.A=1 TO V-1
215 MOVE((D?A)+90),((C?A)-48)
220 DRAW((G?A)+90),((F?A)-48)
225 N.A
230 REM ii rotation routine ii
250 INPUT $W;%N=0.25;L=1
251 %N=%N+0.43
260 CLEAR4
270 FOR A=1 TO V-1
280 Z=(D?A)-48;X=(B?A)-128
281 Y=(C?A)-144;GOS.c
290 MOVE (128+K),(96+Q)
300 Z=(G?A)-48;Y=(F?A)-144
301 X=(E?A)-128;GOS.c
310 DRAW (128+K),(96+Q)
350 NEXT A
352 INPUT $W
354 IF $W="" ;G.251
357 IF $W="X";L=2;G.251
360 IF $W="R";GOTO e
361 END
370 REMiiiiii rotating iiiii
380c IF L=2;%Z=Z;K=X;G.y
382 %T=ATN(Z/X);%I=Z/SIN(%T)
385 %K=COS(%N+%T)*%I
387 %Z=SIN(%N+%T)*%I;K=%K
390y %T=ATN(Y/%Z);%I=Y/SIN(%T)
393 %L=COS(%N+%T)*%I;Q=-(%L);R.
470 P.$12
471 REM the start of the prog
475 P."         ORTHOGRAPHIC TO"'
477 P."         *3D* CONVERTER"''
480 PRINT"   choose mode"'
490 INPUT"'R'=RECORD;'P'=PLAYBACK"$W
500 IF $W="R";R.
505 REM tape file input
510 PRINT"   playback"'
512 FOR J=1 TO 40;WAIT;WAIT;N.J
515 PRINT"POSITION TAPE ON FILE LEADER"'
520 PRINT"play AND PRESS A KEY"'
525 LINK £FFE3
530 V=BGET X
535 PRINT"file found"'
540 FOR Y=1 TO V
550 B?Y=BGET X;C?Y=BGET X;D?Y=BGET X
```

```
555 E?Y=BGET X;F?Y=BGET X;G?Y=BGET X
560 NEXT Y
570 PRINT" loaded;PRESS A KEY"'
575 LINK £FFE3
577 CLEAR4
580 FOR Y=1 TO V
590 MOVE (B?Y),(C?Y);DRAW (E?Y),(F?Y)
595 MOVE (B?Y),(D?Y);DRAW (E?Y),(G?Y)
600 NEXT Y
602 V=V+1
605 INPUT $W;G.205
615 REM ii recording data ii
620e CLEAR 0
630 PRINT"  record object"'
632 F.J=1 TO 40;WAIT;WAIT;N.J
635 PRINT"PUT TAPE INTO record AND ALLLOW"'
640 PRINT"A LONG LEADER THEN PRESS A KEY"'
645 V=V-1
650 LINK £FFE3
660 BPUT X,V
665 WAIT;WAIT;WAIT;WAIT;WAIT
670 FOR Y=1 TO V
680 WAIT;WAIT;WAIT;BPUT X,(B?Y)
685 WAIT;BPUT X,(C?Y);WAIT;BPUT X,(D?Y)
690 WAIT;BPUT X,(E?Y);WAIT;BPUT X,(F?Y)
695 WAIT;BPUT X,(G?Y)
700 NEXT Y
710 PRINT" file recorded"'
720 PRINT"    PROGRAM FINISHED";END
730 REM iiii bug killer iiii
735n F.A=1 TO V-1
740 IF D?A=48;D?A=49
745 IF B?A=128;B?A=129
750 IF G?A=48;G?A=49
755 IF E?A=128;E?A=129
760 IF C?A=144;C?A=145
765 IF F?A=144;F?A=145
770 N.A;R.
```

From 1st July ACORN are bringing out a monthly magazine called 'ACORN USER'. According to the literature the magazine will include accurate up to the minute technical information on all present and future products, with regular hardware and software reviews. Subscription is £15 per annum including postage.

VARUNA ELECTRONICS are marketing a real time clock and calendar for the ATOM. The unit is battery backed-up and fits inside the ATOM. It provides output in tenths of a second, seconds, minutes, hours, days, months and also features automatic leap-year calculation. More information on this product can be obtained from: VARUNA ELECTRONICS, Horsell Park, Woking, Surrey.

CLEVER COMPUTER CONTROLS have started up a new service for ATOM users. Simply send them a tape with your program on and they will put it into EPROM for you. A number of programs can be held on one EPROM and a directory is utilised to inform the user where everything is. Price is quoted as £7.95 for a 4K EPROM. They are also marketing a board called FLEXIROM at £39.50 which is a 4K CMOS RAM board which has an on-board ni-cad battery. The unit is designed so that it can be programmed like normal RAM or switched to ROM mode where its data will be retained when the power is removed. Finally they also market an 8 way EPROM socket expander which allows the user to select between 8 4K/ROMS/EPROMS/FLEXIROMS. More infomation from Claycount Lower Hardnes, Canterbury, Kent.

DEIMOS are marketing a very versatile EPROM programmer operating under software control for £45.00. A very cheap power supply for the programmer is also available from them. They can also supply a 16K EPROM card and a 16K EPROM/RAM card. More information contact: DEIMOS, Gravely Hill, Eckington Rd., B'ham.

PSION COMPUTERS are marketing a graphics screen dump for the ATOM for GP80 printers as well as the EPSOM, a cassette is available at £3.50. (Unfortunately I do not have the address of this firm.)

JOHN RIGGS unfortunately will be unable to set up a Radio Ham sub-group as indicated in Newsletter 5.

FOR SALE 12K ATOM + Nat. Panasonic tape recorder. Offers to Phillip Jackson, 96 Spalding Avenue, Cliffton, Yorks.

FOR SALE 19K ATOM with disk controller + Shugart SA400 drive, 7A 5V PSU, Wordpack, Database, Utilities, 4 games packs, Peeko-computer, Soft VDU, Maze, Invaders, RS232 interface, games paddles, RGB monitor output, IEEE printer output, + educational software. Offers in the region of £350 to Paul Beverley 0603 610622.

An attempt is being made to set up a computer user group in the ORTON area of Peterborough. Anyone who is interested should contact Roger Tallowin, 31A, Cherry Orton Road, Orton Waterville, Peterborough.

Due to an over-sight we omitted to include the address of the firm who supply a ROM extender card for the ATOM we mentioned in NL/5, it is: C.M.T., 6, Turf Park Road, Royton, Oldham, Lancs.

ECONET CONCLUSIONS

ECONET is a versatile networking system capable of a very good turn of speed. However, the file server appears a little slower than a floppy by itself. As a method of giving each user bulk storage it is nearly as good as the latter. Network security needs looking at. There appears to be only one safeguard in that you cannot *REMOTE the file server. You can however remote any other station and it will not take long in the classroom for some junior genius to find that out.

While the main interest in ECONET must be in the classroom, there is a certain "hobby interest" if local computer clubs start experimenting. ECONET is also suitable for low cost commercial and industrial applications. Indeed the first commercial products using ECONET are beginning to appear for some specialised applications.

RICHARD MEREDITH

SOFTWARE LIBRARY

BIO Find out just when you are at your physical and intellectual peaks. TEXT 4K VDU 6K.

BOMBS AWAY High resolution bomb attack on a profile of a city. TEXT 4K VDU 6K

BREAKOUT Demolish a wall of bricks with a bat and ball, good sound effects.

CONFACTORS A program for giving approximation of gear ratios for gearing in machine tools, and pairs of factors to help in selecting suitable pick-off gears.

CONTACTS Allows names, addresses and phone numbers to be stored away on tape for subsequent retrieval. Ideal for those like me who write phone numbers on the back of envelopes and then throw the envelope away. 12K.

DEBUG A debugger for machine code programs similar to the monitor found on machines like the NASCOM. Written in a hybrid of BASIC and assembler. TEXT 3K.

DEFFEQ Solves second order differential equations and plots their response to a step input in graphics mode 4. Requirements as Spirogra.

DODGEMS Navigate your way through a maze collecting points on the way while being chased.

DOMINOES Play against the ATOM with this version of 5's and 3's. TEXT 4K.

DUCKSHOOT You have disturbed a field of ducks and you have to shoot them before they fly away. TEXT 2K.

EDITOR Allows pages of text to be created and edited ready for outputting to the printer. TEXT 4K.

ETCHA Allows you to "draw" pictures on the screen (like the ETCHA-SKETCH toy). Works in mode 0 TEXT ½K.

FINANCIAL Calculates mortgage repayments, interest on short term loans, etc.. Great for sleepless nights. TEXT 3K.

FIND SNOOPY Find Snoopy lost in a forest using only compass points to direct you.

HWRITE Allows the mixing of high resolution graphics and text, by replacing existing VDU routines with another which creates the dot pattern needed to form the 64 character ASCII subset, as well as recognising most of the ATOM control codes. 12K.

LIFE Very fast version of Conway's game, simulating life the universe and everything. Doesn't take 7½ million years to run either. TEXT 4K.

LUNAR LANDER Land your space craft on a randomly generated landscape which enlarges as you get closer. TEXT 3K VDU 6K.

MAZE Navigate your way out of a maze using 3D projections of its passages to help you. TEXT 4K VDU 6K.

MISSILE Use skill and judgement to eliminate passing ICBM's. TEXT 1K VDU½K.

MOONLANDER Given a certain amount of fuel you have to land your space-craft safely on the moon. TEXT 1½K.

MOUSETRAP Catch a mouse by "drawing" a trap round it as it moves over the screen. TEXT 1.5K VDU mode 1.

OTHELLO Play against the ATOM or a friend using the fast response time game of OTHELLO. TEXT 4K VDU mode 3.

3D OXO Play 3D "O"'s and "X"'s against the ATOM on a 4*4*4 matrix. Tricky except for those hypermaths who play on 4*4*4*4 boards.

SOFTWARE AND HARDWARE LIBRARIES

PACK — A machine language utility for reducing the size of a program by condensing multiple spaces into a single space. This method ensures that syntax errors due to missing significant spaces cannot occur. Also readability is improved although the space saving is less than that obtained by eliminating all spaces. Please note that this program will run on an expanded ATOM.

PASSE-TEMPS — Play against the ATOM in this version of "Connect 4". TEXT 3K.

PONTOON — An ATOM version of this popular game. TEXT 3K VDU 1K.

RENUMBER — A fast renumber routine written in assembly language suitable for all sizes of ATOM.

SERIAL — This is a teletype/VDU driver to replace the inbuilt memory mapped display of the ATOM. Written in assembler, it is suitable for all ATOMS of over 4K which have at least 4K available on a continuous block (for the source code). The binary is less than 256 bytes. Any ATOM with the lower text space in operation can run the binary. Either the display only or the keyboard and display can be replaced. It requires the 6522 VIA and a modicum of external circuitry to drive either an RS232 or loop terminal.

SHARP — One for those with elephantine memories. You have to remember some numbers that have been displayed for a short time. TEXT 1K VDU ½K.

SPEED BOAT — Navigate your way round a lake avoiding various obstacles using appropriate nautical terms. TEXT 4K.

SPIROGRA — Draws spirograph patterns into graphics mode 4. Requires 6K for VDU. Also needs the floating point ROM.

SWART — The software UART routine used to provide serial i/o in the above serial listing, but no control code recognition or operating system interface. Hardware requirements 3K TEXT, VIA, minimal external circuitry.

HARDWARE LIBRARY

TV to video monitor conversion (specifically for a PERDIO TV)

On board memory expansion beyond 12K.

Interfacing the ATOM to a Commodore printer.

Cursor control with individual keys.

Using the ATOM with an EPSOM TX80B printer.

13K static RAM card for the ATOM.

ATOM time trials - all BASIC commands timed.

An analogue voltage interface and joystick controller.

ORDERING LISTINGS

Prices of software and hardware listings are as follows:-

LISTINGS — 20p per program (25p for overseas users).

NEWSLETTERS — Back issues are available for issues (1,2,3,4,5) at 60p each (80p for overseas users).

Cheques should be made payable to "THE ATOM USER GROUP" and sent to the address on the front page.

ATOM USER GROUP

Newsletter 7

Welcome once again to the Atom User Group Newsletter. A bit late I'm afraid, but help in producing the Newsletter is non-existent at the moment. I hope to bring out Newsletters 8 and 9 in the near future to put us back on the right track. Thanks for your patience.

For a change I have removed the Software Library list from this issue to provide more room for articles. If any member would still like to order listings they are still available at the old prices. The complete library contents can be found in Newsletter 6.

Once again thanks to all of those who have contributed to this issue. If anyone out there has anything they think may interest our members please forward the relevant information.

PETER FROST

CONTENTS

USING THE COMMODORE VIC JOYSTICK WITH THE ATOM

The VIC joystick is easy to connect to Atoms' fitted with the VIA chip. It is reasonably priced at under £8 and provides a much better means of input than the usual finger twisting combinations of SHIFT/REPT/CTRL/Cursor keys, it also means that the Atom can be placed safely out of the way when children are playing games.

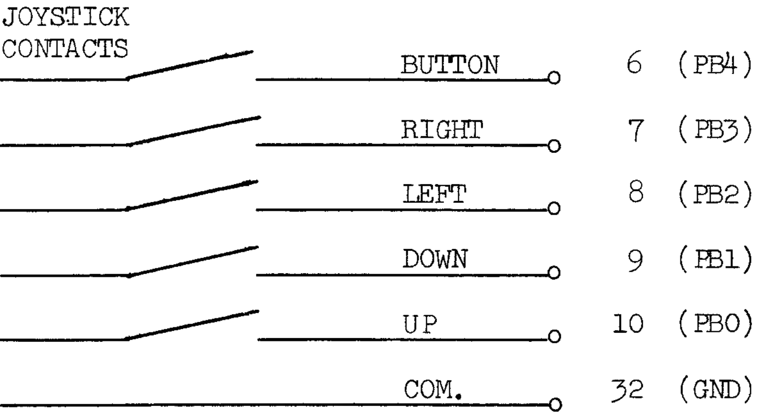
Most commercial games should be modifiable with a bit of detective work and/or phone calls to the manufacturer, perhaps users who manage to do so could pass the word on to others via the User Group.

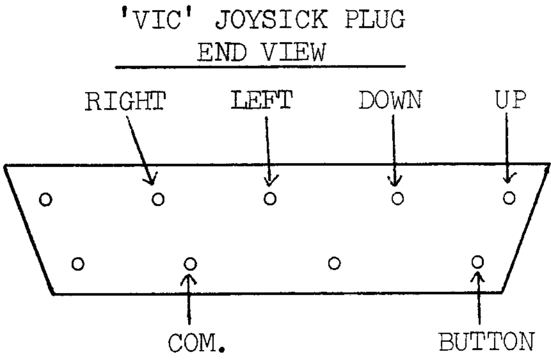
The joystick is the simple non-proportional type consisting of five small switches, one is for the 'Fire' button and four are operated by the joystick lever such that up/down/left/right movements each close one switch, intermediate directions close the 'nearest' pair of switches.

VIA port B is set up for input and the five low bits are used, one for each switch in the joystick. The common line from all switches is returned to ground so that moving the stick or pressing the button pulls one or more bits low (the VIA has internal pull-up resistors). A simple BASIC routine can then sort out which direction the stick is being moved and if the button is depressed. For fast games it will probably be necessary to do the equivalent in machine code.

The socket on the joystick lead requires a 9 way sub-min 'D' type connector which can be connected to the 64 way Euroconnector on the Atom with a short length of 6 way ribbon cable, the diagram gives pin connections.

The example program shows how it works and could be extended to make a simple game of some sort, as it stands a spot moves around the screen under control of the joystick and the button just produces a beep.

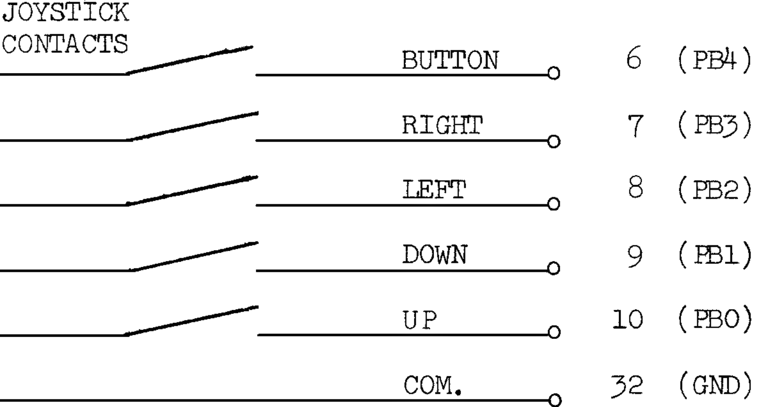




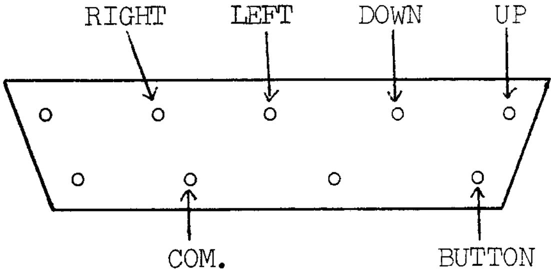
Bus Connector Pins.
Side B Issue 1 Atoms.
Side A All Others.

Test Prog:

```
 10 ?≠≠B802=0;A=≠≠B800;X=64;Y=32       Set up
 20 CLEAR 1
 30 DO
 40 PLOT 13,X,Y                        Plot new spot
 50 H=X;V=Y                            Remember its position
 60 IF ?A 1=0 Y=Y+1                    Up?
 70 IF ?A 2=0 Y=Y-1                    Down?
 80 IF ?A 4=0 X=X-1                    Left?
 90 IF ?A 8=0 X=X+1                    Right?
100 IF ?A 16=0 P.$7                    Button?  Beep
110 WAIT
120 PLOT 15,H,V                        Unplot old spot
130 UNTIL 0
```



'VIC' JOYSTICK PLUG
END VIEW



INVERTED VIDEO AND 8 SIMULTANEOUS COLOURS FOR THE ATOM

The Video Display Generator (VDG) for the Atom (MC6847) has the possibility of some additional display modes not being used in the Atom. By adding some small circuits to the Atom it is possible to obtain these modes.

The following additional modes have been found with help of the Motorola data sheet:

1. Inverted video mode
2. Semi-graphics 4 + text (4 colours)
3. Semi-graphics 4 (8 colours without text)
4. Semi-graphics 6 (4 colours without text)

1. Inverted Video

This mode gives dark letters against white background. The colour of the background is light green. The border is black as before.

The connection between pin 40 and pin 32 of the VDG must be broken and the signal at pin 40 must be inverted and fed to pin 32 in order to obtain the inverted mode. An inverted mode is also obtained by connecting pin 32 to +5v (through a resistor) but then the cursor (which is black) will be missing. Programming can be done by means of one of the spare ports of the VIA (6522, IC1). A circuit which is used to switch from Normal to Inverted mode is shown in Fig. 1. The signal from port PB0 (pin 10 of the VIA, or pin b10 of PL6 or PL7) is used for control. This voltage is normally +5v and saturates the transistor which is connected between pin 40 and pin 32 of the VDG. By typing ?#B802=1 the voltage switches to 0v and Inverted Video is on. It was found that the line sync on some TV sets might fail. In some cases it might help to adjust the video signal level.

2. Semi-graphics 4 + Text

This mode uses an internal "course graphics" generator with a resolution of 64X32 elements in B/W or colour. The text generator can be used and 4 colours can be present on the screen at the same time. Pin 31 (INT/EXT) of the VDG must be disconnected from the circuit to obtain this mode. A transistor switch is used as shown in Fig. 2. We have used port PB1 of the VIA (pin 11) for control. This port is also available at pin 69 of PL6 or PL7.

3. Semi-graphics 4 Without Text

In this mode the internal text generator is switched off. 8 different colours are then available on the screen at once. Pin 34 of the VDG must be disconnected and fed to +5v to obtain this mode. This is done by means of a transistor which is connected to port PB2 of the VIA (pin 12), or pin b8 of PL6 or PL7, as shown in Fig. 2.

4. Semi-graphics 6

The resolution of this mode is 64X48 which is similar to CLEAR 0. Both pin 31 (INT/EXT) and pin 34 (A/S) of the VDG must be disconnected and fed to +5v to obtain this mode. Port PB1 and PB2 of the VIA are used for control as shown in Fig. 2.

As for CLEAR 0 and second set of 4 colours can be obtained by typing: ?#B002=8. The normal set of 4 colours are returned by typing: BREAK or ?#B002=2.

When ports PB0, PB1 and PB2 of the VIA are used as shown in Figs. 1 and 2, the control codes are as follows:

1. Inverted Video Mode : ?#B802=1
2. Semi-graphics 4 + TEXT: ?#B802=2
3. Semi-graphics 4 : ?#B802=6
4. Semi-graphics 6 : ?#B802=4
5. Back to Normal : ?#B802=0 or BREAK

Cont/d....

Transistors BC 239 have been used in the circuit. Similar transistors should also work. The components are mounted on a small board on top of the VDG. Pin 31, 32 and 34 of the VDG have been bent outwards external to the socket and short leads from this board are soldered (with care!) directly to these pins.

For the connections to the VIA we have used the empty holes of PL7 pin b10, b9 and b8. These are located on the fourth row of holes from the edge of the Atom PCB as shown in Fig. 2.

The new semi-graphics modes can be used for poking. For plotting new routines are required. The listing of two such programs are given below. The program SEMI-GRAPHICS 4 provides TEXT by altering line 450 (B802=2). In this case the colours C=4 to C=7 are available.

P.H. and K.L. Grønhaus

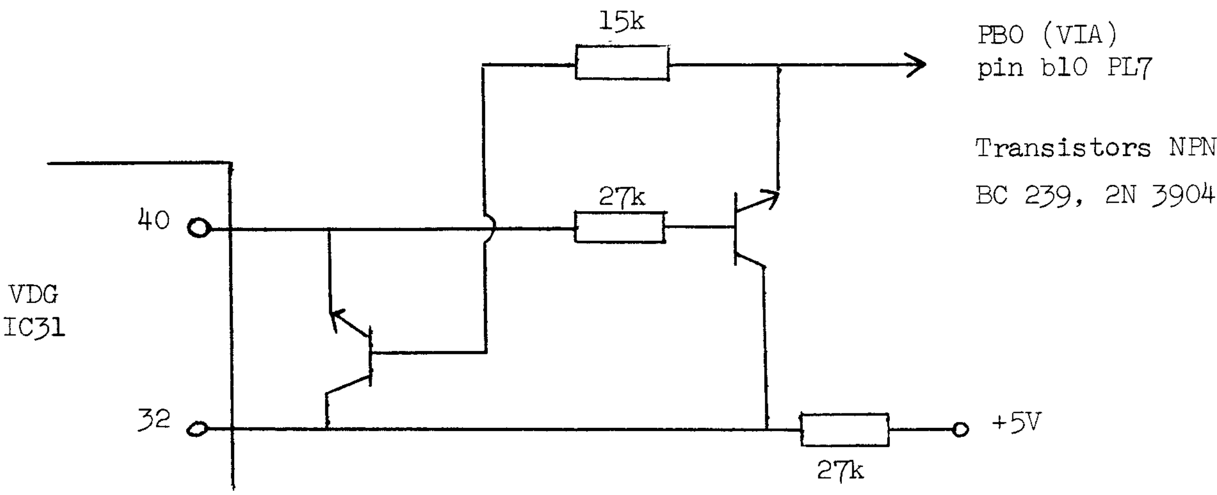


Fig. 1 INVERTED VIDEO

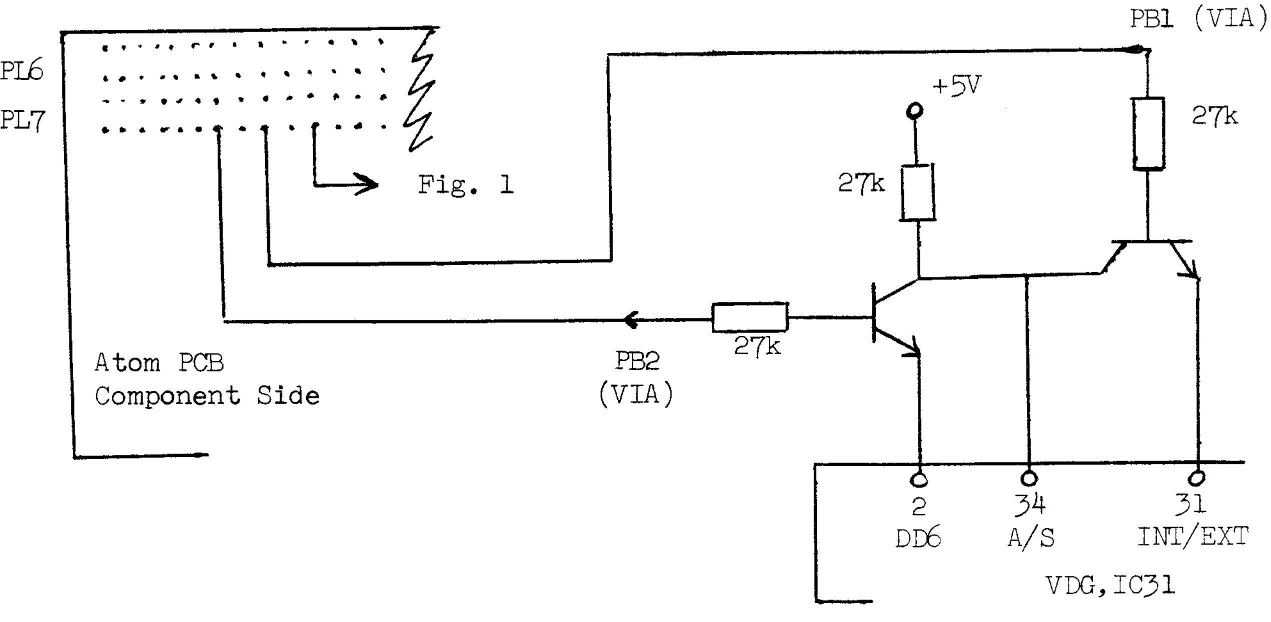


Fig. 2 SEMIGRAPHICS 4 and 6

```
100 REM SEMIGRAPHICS 4 -
110 REM POINT PLOTTING ROUTINE.
120 REM COLOURS :
130 REM C=0 Green
140 REM C=1 Yellow
150 REM C=2 Blue
160 REM C=3 Red
170 REM C=4 Buff
180 REM C=5 Cyan
190 REM C=6 Magenta
200 REM C=7 Orange
210 DIMV7,LL2,Q-1
220 !V=#30201000;V!4=#70605040
230 P.$21;F.I=0TO1;P=Q
240[LDA#5B;ORA#5D;BNELL0
250 LDA#5A;CMP@64;BCSLL0
260 LDX#5C;CPX@32;BCSLL0
270 LDA@#80;STA#60;CPX@16
280 BCSP+4;INC#60;TXA;AND@#E
290 EOR@#F;ASLA
300 ASLA;ASLA;ASLA:STA#5F
310 LDA#5A;LSRA;ORA#5F
320 LDX#5A;CPX@32;BCSP+4
330 AND@#EF;STA#5F
340 LDX#324;LDY@0;LDA(#5F),Y
350 AND@#F;ORAV,X;STA(#5F),Y
360 LDA#5C;LSRA
370 PHP;LDA#5A;LSRA;PHP
380 LDA@0;PLP;ADC@0
390 PLP;BCSP+4:ADC@2
400 TAY;LDA#F7CD,Y;LDX#5E;LDY@0
410 DEX;BEQLL1;DEX;BEQLL2
420 EOR#FF;AND(#5F),Y;STA(#5F),Y;RTS
430:LL1ORA(#5F),Y;STA(#5F),Y;RTS
440 :LL0RTS;];N.;P.$6
450 ?#B802=6;REM SEM.4 MODE
460 CLEAR0;!#3FE=Q
470 REM PLOT DEMONSTRATION
480 DOC=A.R.%8
490 F.I=0TO7
500 WAIT;DRAW(A.R.%64),(A.R.%32)
510 N.;U.0
```

```
100 REM SEMIGRAPHICS 6
110 REM POINT PLOTTING ROUTINE
120 REM COLOURS :
130 REM ?#B002=  2 / 8
140 REM C=0 Green  / Buff
150 REM C=1 Yellow / Cyan
160 REM C=2 Blue   / Magenta
170 REM C=3 Red    / Orange
180 Q=#3000
190 F.I=0TO57;I?Q=I?#F6E2;N.
200 P=Q+58;P.$21
210[TAY;LDA#324;STA#61;LDX@0
220 LDA(#5F,X);ASLA
230 ASLA;LSR#61;RORA;LSR#61
240 RORA;STA(#5F,X);];P.$6
250 F.I=0TO29;P?I=I?#F71D;N.
260 CLEAR0;!#3FE=Q
270 ?#B802=4
280 REM PLOT DEMONSTRATION.
290 DOX=A.R.%64;Y=A.R.%48
300 C=R.;WAIT;DRAWX,Y
310 U.0
```

```
100 REM CHARACTER
110 REM DEMONSTRATION.
120 CLEAR0
130 F.I=0TO#FF;I?#8000=I;N.
140 F.I=0TO6;?#B802=I
150 LI.#FFE3;N.
160 ?#B802=0;E.
```

BASIC Subroutines from Machine Language on Acorn Atom

The following utility, packaged as a single line of text, allows any BASIC subroutine to be called from machine language by a single JSR instruction. I find it invaluable during the writing of a program.

To set up the utility, type the following (with [carriage return] after each line as usual):-

```
ØREM;LI.≠≠FCBA;  followed by 43 spaces
P=≠≠291Ø
[PHP;PHA;TXA;PHA;TYA;PHA;LDA6;PHA;LDA5;PHA
[LDA@5;STA5;LDA@≠≠29;STA6;LDY@Ø;LDA≠≠8Ø;JSR≠≠CC25;JSR≠≠CBD5
[PLA;STA5;PLA;STA6;PLA;TAY;PLA;TAX;PLA;PLP;RTS
```

To check that it has been correctly set up, type:

```
T=Ø;I=≠≠29Ø1;DO T=T+?I;I+1;U.?I=13;P.T
```

This should give the answer 4949.

Once the utility has been set up, any BASIC subroutine with a label can be called from any point in a machine language program by putting the ascii code of the label into the address ≠≠8Ø and then executing JSR≠≠291Ø. For example if ?≠≠8Ø=CH"a"=≠≠61 then JSR≠≠291Ø executes the BASIC subroutine GOSUBa.

The utility has the following advantages over the more rudimentary version suggested at page 50 of Acorn Atom Magic Book.

1. It is easy and flexible to use (very necessary during program development).
2. It copies with a BASIC subroutine in ordinary syntax, ending with a RETURN rather than a LINK statement.
3. It leaves A, X, Y, P and the hardware stack as it finds them.
4. It leaves addresses 5 and 6 as it finds them, so that when eventually the machine language program itself terminates, control is returned to the interpreter at the correct place.

As the utility is held directly in machine code it does not have to be reassembled every time a line program is altered. As it is disguised as a line of text it can be SAVEd and reLOADed with the rest of the program. It's weird LISTing can be suppressed, if desired, by ?≠≠29ØØ=Ø (though NEW, and hence Break, reveal it again).

The following is a brief description of how it works:-

1. It pushes the registers and the text pointer addresses 6 and 5 onto the stack.
2. It calls the ROM subroutine JSR≠≠CC25, which finds the address of the label in A and puts it into addresses ≠≠58, ≠≠59.
3. It hitches onto the part of the interpreter responsible for GOSUB at ≠≠CBD5, and the interpreter takes over.
4. The RETURN at the end of the GOSUB returns the interpreter to the statement LI.≠≠FCBA at ~~8~~≠≠29Ø7.
5. The part of the interpreter responsible for LINK pushes ≠≠C3C3 onto the stack as always.
6. The routine at ≠≠FCBA is PLA;PLP;RTS, which clears the unwanted ≠≠C3C3 from the stack and returns to the utility program.
7. Finally it restores addresses 5 and 6 and the registers.

Eric P. Smith

NUMBER BASE CONVERSION

This program prompts for an INPUT BASE which may take the range 2 to 36 inclusive. As a shorthand, the program accepts H or # to mean 16 and also to mean 10. Using letters A to G inclusive are special cases: the input will be taken as the contents of the BASIC variable A-G as appropriate.

The input value is requested (unless INPUT BASE was A-G) and may be in the positive range accepted by BASIC (Ø - 2147483647 decimal inclusive).

Next, the OUTPUT BASE is requested, same rules as for INPUT BASE. If A-G is specified the result is placed in the appropriate BASIC variable. Otherwise the result is printed (preceded by an = sign).

The program is a continuous loop but may be terminated by typing just a RETURN as a reply to any prompt.

By specifying A-G instead of a base, the number can be manipulated by exiting the program and using intermediate-mode commands; final base conversion can be achieved by re-entering the program.

Source code: 1657 bytes.
Arrays: 132 bytes.

G. Manning

```
 1 REM GLM NUMBER BASE CONVERSION 25/9/82.
 2 REM BASE 2-36 (OR H OR # FOR HEX, OR . FOR DECIMAL)
 3 REM OR BASE A-G PUTS VALUE IN INTERNAL REGISTER A-G
 10 DIM Y(60),X(70)
 20 $Y="INPUT BASE"; PRINT '
 30 GOSUB a; IF Z=0 END
 40 GOSUB d; IF Z=0 END
 50 $Y="OUTPUT BASE"
 60 GOSUB a; IF Z=0 END
 70 GOSUB c
 80 GOTO 20
100
110 ***************
120 * SUBROUTINES *
130 ***************
140
150aREM GET BASE NUMBER
160 PRINT $Y; INPUT $X
170 IF LEN(X)=0 THEN Z=0; RETURN
175 IF LEN(X)>2 GOTO a
180 IF LEN(X)>1 GOTO t
190 IF ?X>49 AND ?X<58 THEN Z=?X-48; RETURN
200 IF ?X>64 AND ?X<72 THEN Z=?X+35; RETURN
210 IF ?X=72 OR ?X=35 THEN Z=16; RETURN
215 IF ?X=46 THEN Z=10; RETURN
220tIF ?X<49 OR ?X>51 OR X?1<48 OR X?1>57 GOTO a
230 Z=(X?1-48)+(10*(?X-48))
240 IF Z>36 GOTO a
250 RETURN
260
```

```
300dREM CONVERT TO INTERNAL REPRESENTATION IN W
310 IF Z>99 GOTO (300+Z)
320 INPUT "VALUE",$X
330 IF LEN(X)=0 THEN Z=0; RETURN
340 FOR V=0 TO (LEN(X)-1)
345 IF X?V>47 AND X?V<58 THEN X?V=X?V-48; GOTO k
350 IF X?V>64 AND X?V<91 THEN X?V=X?V-55; GOTO k
355eV=(LEN(X)-1); NEXT V; GOTO d
360kIF X?V>=Z GOTO e
365 NEXT V
370 U=1; W=0
375 FOR V=(LEN(X)-1) TO 0 STEP -1
380 W=(U*(X?V))+W
382 IF U=0 GOTO 384
383 IF W/U=(X?V) GOTO 385
384 V=0; NEXT V; GOTO d
385 U=U*Z
390 NEXT V
395 RETURN
400 W=A; R.
401 W=B; R.
402 W=C; R.
403 W=D; R.
404 W=E; R.
405 W=F; R.
406 W=G; R.
410
500cREM OUTPUT CONVERSION
510 IF Z>99 GOTO (500+Z)
520 V=0; IF W<0 GOTO 590
530 DO
535 X?V=W%Z
540 W=W/Z
545 IF X?V<10 THEN X?V=X?V+48
550 IF X?V>9 AND X?V<48 THEN X?V=X?V+55
555 V=V+1
560 UNTIL W=0
565 PRINT "=";
570 FOR U=(V-1) TO 0 STEP -1
575 PRINT $X?U
580 NEXT U
590 PRINT '; RETURN
600 A=W; R.
601 B=W; R.
602 C=W; R.
603 D=W; R.
604 E=W; R.
605 F=W; R.
606 G=W; R.
```

THE ACORN ATOM DISK PACK

The pack includes a Disk Drive with a built in power supply and fitted plugs, all the connecting cables, a utilities Disk, and a booklet explaining what had to be done to the Atom and how to operate the disk pack. This was as expected, clear and easy to understand. ICs 2-5 and the 64-way connecter PL6 must be fitted, costing around £6.20, but the disk pack also provides an extra 1K byte of RAM from #3C00 to #3FFF thus giving 13K to previously expanded Atoms.

Filenames do not need to be in quotes with DOS commands such as *SAVE unless they contain spaces or quotes - a great saving in time and effort. Each file has a 'Qualifier', which can be any alphanumeric character, and DOS only accesses files with the current qualifier. It is therefore possible to have several files with the same name but different qualifiers.

*DOS — Enters DOS. The COS is re-entered on BREAK.

*CAT — Catalogues a disk from the disk, or if it is still running from the catalogue buffer at #0200.

*DIR — Copies the catalogue from disk into the buffer.

*TITLE HELLO — Sets the title of the disk to 'HELLO'.

*SET A — Sets the qualifier to 'A'.

*USE A — Sets the qualifier to 'A' only for the next successful operation.

*LOCK FRED — Locks file 'FRED', preventing deletion or re-saving.

*UNLOCK FRED — Unlocks file 'FRED'.

*INFO FRED — Displays information about file 'FRED'.

Qualifier:	lock state	filename	load address	execution address	length (bytes)	start sector
e.g.	a:	FRED	2900	C282	00300	005

#FRED indicates the file is locked.

*MON — Following operations on a file print out information as *INFO.

*NOMON — Turns off *MON.

*SAVE FRED — As in AT&P.

*LOAD FRED — As in AT&P.

*DELETE FRED — Deletes file 'FRED' from the disk.

*GO 2800 — Same as 'LINK #2800'.

*RUN "FRED" "HELLO" — Same as in AT&P but puts 'HELLO' in memory at #0140 where it can be accessed by the user's routine.

*EXEC FRED — Executes text file 'FRED' as if it had been entered from the keyboard.

Files can be accessed using PUT, GET, FIN, etc. as in AT&P.

The utilities on the utilities disk are:

FORMAT — Initialises a disk, prepares it for use and verifies each sector.

COMPACT — 'Tidies up' a disk, thus increasing loading and saving speeds, providing more space and displays the number of free sectors.

INFALL — Does *INFO on all files on a disk.

VERIFY — SEARCHES for any sectors which contain errors.

D. Poirier

ATOM USER GROUP

Newsletter 8

Hello once again. Well the computing scene appears to be maintaining its rapid advances in CPU, memory, and peripheral technology. It seems almost impossible to feel totally up to date with the current state of the art. However, one particular area which is taking an uncomfortably long time to arrive is cheap mass storage for the hobby user. Cassette storage is simply not up to the task as software becomes more complex and necessarily longer. Systems themselves have got progressively more sophisticated but in the end the old bottleneck of loading and saving software keeps arising. Two recent developments may provide the ray of hope we have all been waiting for. The first is the micro drive. This device is effectively a small but ruggedised version of the floppy, but promises to offer the same storage capacity at half the cost. In fact these devices have been around a while but for some reason no entrepreneur has come along and satisfied the potentially huge demand amongst hobby users. The second is the advances being made in the area of non volatile RAM. These beasties operate just like normal RAM and at comparable speeds, but retain their information when the power is removed. 64K bit devices are now available, and whilst they are still expensive, I don't think it will be long before we can find this kind of memory being used to hold utility software and favourite games inside home micros.

Inside this Issue an article dealing with the possibility of interfacing your ATOM to receive PRESTEL is covered. This is a very exciting prospect and if anybody out there has any experience of this I would be very interested to hear from them.

PETER FROST

CONTENTS

INTERFACING THE ATOM TO THE COMMODORE PRINTER

The Commodore printer uses the so called IEEE 488 general purpose interface bus which is a complex system by which a number of different devices can all be connected together. In order to drive the printer we have to make use of both ports on the 6522 VIA (versatile interface adapter).

As with the normal parallel printer interface on the ATOM, port A of the VIA is used to output the data, but then part of port B has to be used for control and management, PB0 and PB1 as outputs and PB6 and PB7 as inputs. Bits 0 to 6 of port A are available as buffered outputs on the printer connector of the ATOM, PL5. Unfortunately the eighth buffer in the 74LS244 octal buffer is used for the CA2 control line. Therefore to make full use of the Pet printer, including its graphics characters, you will need to take the PA7 line straight from the 6522 VIA on PL6. This of course is not buffered, but provided you do not use excessively long cables, that should not cause a problem. Figure 1 gives the connections on the IEEE connector and the lines on the ATOM to which they should be connected.

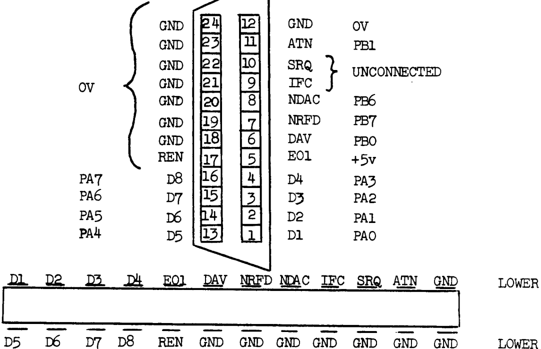
From the software point of view the best way to operate the printer is to use a machine code routine. (See NL 6 for information on how this is done.) The software is shown below and although it is fairly well documented you really will need to refer to the article in Newsletter 6 to really understand what is going on. There are certain areas of the routine which are peculiar to the Pet printer, and in order to explain these we will first have to look at the unusual arrangement of the Pet's character set. These problems have been overcome in the software but essentially are:

1. Upper and lower case characters are both produced by using the same ASCII codes (65-90).

2. When an odd number of quotation marks has been sent to the printer all control codes are not acted on but printed out as inverted characters.

The only real problem found after implementing the changes is that in the solution to problem 2 the software prints out two quotation marks instead of one. Control codes can be sent out to the printer by using the PRINT$ format. The effect of P$1 is that succeeding characters on the line will be enhanced i.e. instead of being in a matrix 7*6 dots the width is increased to 12 dots. To revert to normal you have to type P.$129.

TIM EDWARDS



Program A - Pet Printer Interface Routine

```
 10 B=#B80C;V=#208;L=11
 20 DIMLLL;FORN=0TOL;LLN=#FFFF;N.
 30 I=#3B26;M=I+#E5
 40 P.$21;FORN=1TO2;P=I;[
 50:LL0 LDAV;STAM+2;LDAV+1;STAM+3\remove old WRCVEC
 60      LDA@(LL2%256);STAV;LDA@(LL2/256);STAV+1\new WRCVEC
 70      LDA@#FF;STAB+3;LDA@3;STAB+2\set up VIA outputs
 80      LDA@0;STAM+1\printer off
 90      LDA@#24;LDX@1;JSRLL1\make printer listen
100      LDA@13;LDX@3;JSRLL1\send a carriage return
110      RTS
120:LL1 EOR@#FF;STXB;STAB+1\output a byte of data
130      BITB;BPLP-3\wait for READY
140      DEX;STXB\ set DAV true
150      BITB;BVCP-3\wait for DATA ACCEPTED
160      INX;STXB\ reset DAV
170      RTS
180:LL2 PHP;STAM;TXA;PHA;TYA;PHA\entry point from WRCVEC
190      JSRLL3\deal with the character
200      PLA;TAY;PLA;TAX;LDAM;PLP;JMP(M+2)\restore & return
210:LL3 LDAM;CMP@2;BNELL4\ctrl b?
220      LDA@128;STAM+1\switch on printer
230      LDA@27;STAM;RTS\return with esc
240:LL4 CMP@3;BNELL5\ctrl c?
250      LDA@27;STAM+1;STAM\switch off printer\ return esc
260:LL5 CMP#FE;BNELL6;RTS\character not sent to printer
270:LL6 BITM+1;BPLLL11\printer on?
280      LDX@3\set ATN high
290      LDAM;CMP@13;BNELL7\carriage return?
300      LDY@0;STYM+4\switch to u.case
310:LL7 AND@#60;CMP@#60;BNELL9\1.case character?
320      BIT M+4;BMILL8\1.case ON?
330      LDA@17;JSRLL1;LDA@128;STAM+4\switch to 1.case
340:LL8 LDAM;AND@#DF;JSRLL1;RTS\output 1.case character
350:LL9 BITM+4;BPLLL10\1.case ON?
360      LDA@145;JSRLL1;LDA@0;STAM+4\switch back to u.case
370:LL10LDAM;JSRLL1\output u.case character
380      LDAM;CMP@34;BNELL11;JSRLL1\double quotation marks
390:LL11RTS
400];N.;LINKI
410 P.$6$2"INITIALISE"&I'"BYTE O/P"&LL1'
420 @=5;P.$3" *SAVE PET "&I,&P,&I';E.
```

Using the programmable graphics character.

```
 10 @=5
 20 PRINT $2$21"    &     #    NML  $17  $18 $17$18"
 30 PRINT "     &     #    NML  $17  $18 $17$18"''
 40 FOR J=32 TO 95
 50 IF J<>34 N=J;GOS.a;G.80
 60 PRINT J,&J"    "$J"    "$17$J"    "
 70 PRINT $145$18$J"    "$17$J
 80 PRINT $145$146"          "
 90 N=J+128;GOS.a
100 PRINT'
110 NEXT J
120 PRINT $3$6
130 END
200aPRINT N,&N"     "$N"     "$17$N"     "$145$18$N""
210 PRINT$17$N
220 RETURN
```

To print out the entire Pet character set.

```
 10 Q=#3B5A
 20 X=1;A=#3F;LINK Q
 30 A=#24;LINK Q
 40 A=#65;LINK Q
 50 X=3;A=20;LINK Q
 60 A=54;LINK Q
 70 A=118;LINK Q
 80 A=118;LINK Q
 90 A=55;LINK Q
100 A=20;LINK Q
110 X=1;A=#3F;LINK Q
120 A=#24;LINK Q
130 X=3;A=254
140 P.$2$21'"Acorns Rule OK?
150 FOR N=1 TO 400
160 LINK Q
170 P." "
180 NEXT N
190 P.'$3$6
200 END
```

NEWSLETTER CONTENTS 1-7

Due to popular requests here is a list of topics covered by the first seven newsletters produced by the Group. Back issues are available at 60p each (80p for foreign users).

Newsletter 1

TV interface, some problems.
Operator precedance (?! :+-/).
Printing strings from assembly language.
Efficient use of memory.
Backus Naur Form.
A fast renumber.
Loops and exiting from them.

Newsletter 2

Atom power supply for extended Atoms.
Text storage (how it is done).
Keyboard input routines.
Useful addresses in the Atom.
Serial printer interface.

Newsletter 3

Cassette recorders (getting them to work).
Loading hex into the Atom.
Useful addresses in the Atom ROM.
Keyboard input routines (again).
Info on faults in the Atom manual.

Newsletter 4

Use of Willow utility ROM.
More useful addresses in the ROM.
Review of the Acorn word pack.
Manchester User Group report.
Noise free graphics.

Newsletter 5

Writing special printer routines.
Pulse generator for the Atom.
Data transfer on the Atom.
Timer for the Atom.
Review of Program Powers Toolbox.
BIO-rhythm program listing.
General news of user groups and software.

Newsletter 6

Econet network for the Atom.
Keyboard debounce routine.
Inputting eqns directly into BASIC.
Cheap and simple light pen.
Fast Atom (2 MHz).
Utilising 2 ROMS in the utility socket.
Graphics characters on a white screen.
3D projection program (allows rotation).

Newsletter 7

Using the Commodore joystick on the Atom.
Inverted Video and 8 simultaneous colours.
Basic sub-routines from machine language.
Number base conversion program.
Acorn Atom Disk Pack an overview.

Interesting Feature of the WORDPACK ROM

For those of you lucky enough to have the WORDPACK ROM installed in your Atom, the following information submitted by Mr. E. Normington, will be of interest.

The following routine allows the use of characters with high resolution graphics provided you have the ROM installed.

```
 5   DIM P-1;C=P
10   [LDX & 60;STA E7;LDY &3
20   JSR ACDE; CHANGE READ/WRITE CHARACTER VECTORS
30   JSR AC4B; CLEAR SCREEN.  SET GRAPHICS MODE 4
40   ];LINK C
50   END
```

This routine sets graphic mode 4 and installs the wordpack character generator. This now allows 21 lines of 32 characters. All the Atom control characters work correctly. To position the character cursor at position X,Y use the following code.

```
1000  a ? 93=Y
1010  P=! 96 & FFF+ 100+? EO; ?P=?P;? El
1020  P=8000+Y* 100;! 94=P+ 10000*P+X
1030  ? EO=X; RETURN
```

This puts line 0 at the top of the screen and column 0 on the left. You can now draw pictures and label them properly.

The AMBER 2400 Printer

When using the AMBER 2400 printer with the Atom it can cause problems since some of the graphics data are control codes. However an entry point in ROM at FF10 skips the control codes check for printer in mode 4. The following routine sends the printer all the codes to print a screen up to the Y axis value 144.

```
110  P. 21; REM TURN OFF SCREEN OUTPUT
120  P. 2 ; REM TURN ON PRINTER
130  F.Y=8000 TO 97FF STEP 32
140  P. 17; REM PRINTER CODE FOR GRAPHICS LINE
150  F.X=0 TO 17; A=?(X+Y); REM BYTE FOR PRINTER IN A? ACCUMULATOR
                            LOADS FROM A ON A JUMP TO MACHINE CODE
160  LINK  FF10; REM OUTPUT TO PRINTER
170  N.X; REM NEXT BYTE ON LINE
180  N.Y; REM NEXT LINE
190  P. 3; REM OFF PRINTER
200  P. 6; REM ON SCREEN
```

Note: The Variable A must be used.

C.N. SAYER

NEWS.....NEWS.....NEWS.....NEWS.....NEWS.....

ACORN STAGE MAJOR EXHIBITION FOR FAMILY USE

Acorn computers are to host a major exhibition in August with the emphasis very firmly on family and educational computing. Organised by Computermarket Place Ltd. in conjunction with Acorn Computers and Acorn User Magazine, the show takes place at the CUNARD INTERNATIONAL HOTEL, Hammersmith Broadway, London, between August 25-28.

ACORN USER MAGAZINE

The Acorn User Magazine is interested in any articles members may wish to submit. They can cover any aspect of computing related to the ATOM or BEEB machines. Items should be submitted to Tony Quinn, ADDISON-WESLEY PUBLISHERS LTD., 53 Bedford Square, London.

BOOKS OF INTEREST TO THE BEGINNER

The Good Computing Book for Beginners - reference book and down to earth guide to computing. Whether for use in the school or office or home it explains all you need to know about computing in easy to understand language (well, that is what the publishers say). Price seems reasonable at £2.95.

Before we delve into details about what the enhanced BASIC provides, there first follows a history lesson.

About 2 years ago the BBC decided that it was about time people knew a bit about computers. They had done this before of course with their Horizon programme, "The Chips are Down", which really was a classic, but the planned series intended to deal with the more detailed aspects of programming. For this then the BBC had to choose a computer, and the question was which one. Looking around they quickly realised that for a tutorial programme standardisation was essential, and it would be good politics to use a British machine. Originally the machine they selected was the Newbury Laboratories Newbrain. However, production difficulties caused delays and eventually Newbury sold their interests to Grundy and the BEEB was left computerless. At this point there were three choices: either use and American machine (the VIC), the ZX80 (or ZX81 which was on the horizon) or the ATOM. The BBC really wanted a reasonably cheap machine (no Research Machines despite howls of anguish from schools who already had them), no politics (i.e. Americans or Orientals), but versatile and expandable (howls from Clive Sinclair). During the gestation period ideas grew and a specification for a brand new machine, to be designed by ACORN, was drawn up. This specification which was at the time seen as being extremely ambitious, eventually became the BBC micro.

Two things stand out about the BBC micro design, firstly its BASIC interpreter, which is a very much expanded version of standard BASIC, unlike the ATOM BASIC which is definitely non-standard. Secondly, a graphics resolution four times that of the ATOM. Of course there were other enhancements in the form of sound generators disc controller, network interface, tube, etc., all on the same board.

As the BBC and ATOM both use the 6502 processor, and because ACORN write well structured software, so that hardware dependent routines are called via well defined procedures if the correct environment can be provided, BBC BASIC will run on any 6502 machine, which of course includes the ATOM.

BBC BASIC Card

As the memory map of the ATOM is not compatible with the BBC micro some mods have to be made to ensure compatibility. These are ingeniously taken care of by providing a board which contains the BBC BASIC ROM, an additional 2K of RAM and an EPROM which provides the BBC BASIC with its routines for controlling the ATOM's hardware and several decoders, latches and multiplexers. It connects to the ATOM by removing 4 IC's from the ATOM PCB and plugging in long leads from the BBC PCB. In addition 3 soldered links have to be made. Fitting it is not difficult and is well described in the manual (of which more later) but if you are not sure, take it to your dealer. In addition to the extra board, you must fit the VIA if it is not already fitted. This is used for the timer function in BBC BASIC.

For some reason the quality of the board is not up to ACORN's usual standard, with a complete lack of solder resist. Any purchaser would be well advised to carefully look at the board for any tracks which have shorted together with blobs of solder.

BASIC Interpreter

This proved to be very satisfactory provided you are not expecting the BBC hardware. Most of the BBC functions are included except those dependent on the BBC hardware. For instance the sound statements and *FX calls are unavailable. I do not intend to go on at length about the functions offered by the BASIC as umpteen magazines have covered these in detail. Remember that ATOM BASIC occupies 8K (excluding floating point) with the BBC BASIC being 20K!

Interestingly, fitting the BBC BASIC does not affect the running of ATOM BASIC, plus any wordpacks/utility ROMS etc. you may have. While you can wire it up so you can only use BBC BASIC, you can also wire it up so that pressing CTRL and BREAK will select BBC BASIC and SHIFT BREAK ATOM BASIC. BREAK alone does what is usually does without changing mode. It is not possible however to change from BBC to ATOM BASIC under software control.

BBC Cassette System

The OS functions with the BBC are much the same as the ATOM, i.e. *CAT,*SAVE,*LOAD. The machine code routines are totally different - gone are OSLOAD, OSASCI etc. - in are OSGBPB (even if it doesn't do anything), OSBPUT, OSBGET (much the same as the ATOM although the addresses are different),. OSARGS (doesn't do anything on the ATOM but does on the BEEB), OSWRCH (as ATOM), OSBYTE and OSWORD and OSCLI (same function but different in detail to the ATOM).

There is partial capability between the BBC and ATOM cassette formats. The BBC/ATOM can load ATOM data files using the BGET and BPUT commands but cannot load files created using *SAVE and *LOAD, and BBC/ATOM cannot load BBC micro tapes at all.

The Manual

So far the BBC extension has been seen to be a very healthy piece of kit well worth having, but now we come to the manual. This is quite easily the worst manual I have seen in my life. The real BBC BASIC manual is about an inch thick, with most of it being given over to the description of BBC BASIC. This one is only 38 pages long. Descriptions of each function are terse to the point of disappearing altogether, e.g.

DIV A binary operation giving inter division.

or TIME A pseudo variable which reads and sets the lower 4 bytes of the computational real time clock.

In a few instances details are given of commands which are not available on the ATOM, e.g., VDU 28,0,31,32,0 REM define window.

The define window commands are not available on the ATOM. The beginner would be well advised to buy the BBC manual to help with command explanations.

Other ACORN products

1. DISCS The BBC/ATOM cannot use the ATOM Disk Pack at all in BBC mode, although there is no problem apparently in ATOM BASIC mode.

2. ECONET Contrary to what the manuals have to say, the ATOM ECONET board cannot be even fitted into the same machine as the BBC BASIC. The ECONET board generates an interrupt on reset, which the BBC board cannot recognise or clear, the result is that the interpreter crashes.

3. COLOUR BOARD The BBC BASIC does not support the colour board unfortunately. So sorry if you want BBC BASIC and anything other than cassette, you will have to go and buy a BBC micro.

Overall Impression

This really is a very good addon for the ATOM, which should extend its 'life' considerably.

Hardware Quality	Fair
Ease of fitting	Easy (if experienced)
Quality of language	Very Good
Expandability	Poor
Manual	Dreadful
Value for money	At £50 (approx.) Good

PRESTEL - AN EXCITING APPLICATION FOR YOUR MICRO

Okay hands up all those who would like access to 250,000 pages of information, hands down all those who have linked their ATOM to the PRESTEL service. Hump.... there seems to be an awful lot of you looking rather silly with your hands in the air. Seriously though PRESTEL launched a 300 baud service for microcomputer and teletype terminal access in January of this year. The new service is initially available on PRESTELS London computers (DRYDEN and KIPLING) and will allow connection from any microcomputer or teletype terminal using a 300 baud modem (currently approved by British Telecom).

As mentioned above, the full data base contains about 250,000 pages of information, for example:-

General Information:	News, weather, amusements, buying, booking, travel, sport, etc.
Business Information:	Stocks and Shares, Commodities, Securities, Exchange rates, etc.
Local Information:	Town and region.
Teleshopping:	Also other interactive services.
Alladin's Cave:	Programs for micros that can be down loaded direct into your m/c.
MICRONET 800:	Coming soon a completely new microcomputer data base on PRESTEL specifically designed for microcomputer users.

PRESTEL Charges (excluding telephone call)

Subscription:	Business User	£15.00 per quarter
	Residential User (that's us)	£ 5.00 " "

Time Charge

As a special incentive to the residential user, PRESTEL will not be making any time charge when the system is used in the evenings, Saturday afternoons and Sundays. At all other times the charge is 5p per minute. If anybody out there is interested in subscribing to this service they should contact Andy Anderson, PRESTEL 300 Service, PRESTEL Headquarters, Room 419, Telephone House, Temple Ave., London. He would like to know your name, address, telephone number and type of micro you wish to use.

There are a few technical points that need considering, one of the main ones being that it will be necessary for individuals to write a program to convert from PRESTEL format to ATOM format. The PRESTEL service will publish these free of charge as they become available for different micros. The PRESTEL display is 40*25 which is clearly incompatible with the ATOMS 32*16 display, however scrolling techniques may help. I think a little bit of ingenuity is needed here to provide an acceptable display.

The PRESTEL character set is not commonly used by micros (BBC micro apart) so the 300 baud service will be transmitting a modified set of just ASCII characters. In place of any PRESTEL graphics character, normally the ASCII character * will be sent.

Telesoftware

The Council for Education Technology (CET) have produced a format whereby programs can be held on PRESTEL frames for down-loading onto micros. Several hundred programs are already available on the system and this will be greatly expanded now MICRONET 800 has been added. Each type of micro will require different software to down-load programs and a number of individuals are busy writing these for PRESTEL. If you wish to write your own down-loader then contact Mike Brown, CET Videotext Systems Development Officer, CET, PRESTEL Unit, Burleigh Centre, Wellfield Road, Hatfield, Herts.

VIEWFAX 258

Viewfax 258 was launched on PRESTEL last year and their entire data base revolves around the world of micros and telesoftware. Viewfax 258 is accessible to all PRESTEL users and no additional fees have to be paid. The data base will be providing information on computer clubs, groups and dealers in the West Midlands. This new section of the data base will be primarily aimed at around 10,000 users in the area who will subscribe to PRESTEL as a result of a new Department of Industry project called 'Club 403'. This scheme will provide homes in the West Midlands with PRESTEL sets at greatly reduced costs.

Where does the Information come from?

The $^1/4$ million pages of information on PRESTEL is supplied and updated by about 180 companies (like Viewfax 258), which are called information providers. Other information providers are British Rail, Financial Times, Meterological Office, etc. PRESTEL at the moment has over 22,000 subscribers, most of whom can access the service at local call rates (remember it depends where in the country the computer you are accessing is in relation to yourself). During 1983 it is expected that this number will increase to 50,000. Some information providers such as Mediatel and Lawtel charge an annual fee. However, Mediatel is only available to members of the medical profession. A number of information providers also sub-let pages within their data base to advertisers wishing to communicate their news and information to PRESTEL users.

As I am sure you will agree this is an area of computing that is really going to take off. Clearly to get yourself up and running you are going to invest some money in a modem and sort out some software that will provide an acceptable display. The only disappointing aspect of this excellent service in being charged in certain cases fourfold:

(i) Subscription to PRESTEL.

(ii) Cost of telephone call (could well be long distance rate).

(iii) PRESTEL time charge.

(iv) Fees for specialised information.

As stated above, it is possible to remove the need to pay item (iii) if you only use the service at certain times, and most home users will not need to access specialised information. To the hobby user the telephone call will represent the most serious problem since at 300 baud it will take over a minute to transfer one page of information. When System X is introduced it should be possible to transmit data at 64K baud reducing call time to a minimum. (Special lines and modems would be required for this speed.)

If any group member has managed to take advantage of this service and would like to pass on their experiences to other members I would be grateful if they could drop me a line.

PETER FROST

BIO — Find out just when you are at your physical and intellectual peaks. TEXT 4K VDU 6K.

BOMBS AWAY — High resolution bomb attack on a profile of a city. TEXT 4K VDU 6K.

BREAKOUT — Demolish a wall of bricks with a bat and ball, good sound effects.

CONFACTORS — A program for giving approximation of gear ratios for gearing in machine tools, and pairs of factors to help in selecting suitable pick-off gears.

CONTACTS — Allows names, addresses and phone numbers to be stored away on tape for subsequent retrieval. Ideal for those like me who write phone numbers on the back of envelopes and then throw the envelope away. 12K.

DEBUG — A debugger for machine code programs similar to the monitor found on machines like the NASCOM. Written in a hybrid of BASIC and assembler. TEXT 3K.

DEFFEQ — Solves second order differential equations and plots their response to a step input in graphics mode 4. Requirements as Spirogra.

DODGEMS — Navigate your way through a maze collecting points on the way while being chased.

DOMINOES — Play against the ATOM with this version of 5's and 3's. TEXT 4K.

DUCKSHOOT — You have disturbed a field of ducks and you have to shoot them before they fly away. TEXT 2K.

EDITOR — Allows pages of text to be created and edited ready for outputting to the printer. TEXT 4K.

ETCHA — Allows you to "draw" pictures on the screen (like the ETCHA-SKETCH toy). Works in mode 0 TEXT ½K.

FINANCIAL — Calculates mortgage repayments, interest on short term loans, etc.. Great for sleepless nights. TEXT 3K.

FIND SNOOPY — Find Snoopy lost in a forest using only compass points to direct you.

HWRITE — Allows the mixing of high resolution graphics and text, by replacing existing VDU routines with another which creates the dot pattern needed to form the 64 character ASCII subset, as well as recognising most of the ATOM control codes. 12K.

LIFE — Very fast version of Conway's game, simulating life the universe and everything. Doesn't take 7½ million years to run either. TEXT 4K.

LUNAR LANDER — Land your space craft on a randomly generated landscape which enlarges as you get closer. TEXT 3K VDU 6K.

MAZE — Navigate your way out of a maze using 3D projections of its passages to help you. TEXT 4K VDU 6K.

MISSILE — Use skill and judgement to eliminate passing ICBM's. TEXT 1K VDU ½K.

MOONLANDER — Given a certain amount of fuel you have to land your space-craft safely on the moon. TEXT 1½K.

MOUSETRAP — Catch a mouse by "drawing" a trap round it as it moves over the screen. TEXT 1.5K VDU mode 1.

OTHELLO — Play against the ATOM or a friend using the fast response time game of OTHELLO. TEXT 4K VDU mode 3.

3D OXO — Play 3D "O"'s and "X"'s against the ATOM on a 4*4*4 matrix. Tricky except for those hypermaths who play on 4*4*4*4 boards.

PACK A machine language utility for reducing the size of a program by condensing multiple spaces into a single space. This method ensures that syntax errors due to missing significant spaces cannot occur. Also readability is improved although the space saving is less than that obtained by eliminating all spaces. Please note that this program will run on an expanded ATOM.

PASSE-TEMPS Play against the ATOM in this version of "Connect 4". TEXT 3K.

PONTOON An ATOM version of this popular game. TEXT 3K VDU 1K.

RENUMBER A fast renumber routine written in assembly language suitable for all sizes of ATOM.

SERIAL This is a teletype/VDU driver to replace the inbuilt memory mapped display of the ATOM. Written in assembler, it is suitable for all ATOMS of over 4K which have at least 4K available on a continuous block (for the source code). The binary is less than 256 bytes. Any ATOM with the lower text space in operation can run the binary. Either the display only or the keyboard and display can be replaced. It requires the 6522 VIA and a modicum of external circuitry to drive either an RS232 or loop terminal.

SHARP One for those with elephantine memories. You have to remember some numbers that have been displayed for a short time. TEXT 1K VDU ½K.

SPEED BOAT Navigate your way round a lake avoiding various obstacles using appropriate nautical terms. TEXT 4K.

SPIROGRA Draws spirograph patterns into graphics mode 4. Requires 6K for VDU. Also needs the floating point ROM.

SWART The software UART routine used to provide serial i/o in the above serial listing, but no control code recognition or operating system interface. Hardware requirements 3K TEXT, VIA, minimal external circuitry.

HARDWARE LIBRARY

TV to video monitor conversion (specifically for a PERDIO TV)

On board memory expansion beyond 12K.

Interfacing the ATOM to a Commodore printer.

Cursor control with individual keys.

Using the ATOM with an EPSOM TX80B printer.

13K static RAM card for the ATOM.

ATOM time trials - all BASIC commands timed.

An analogue voltage interface and joystick controller.

ORDERING LISTINGS

Prices of software and hardware listings are as follows:-

LISTINGS 20p per program (25p for overseas users).

NEWSLETTERS Back issues are available for issues (1,2,3,4,5) at 60p each (80p for overseas users).

Cheques should be made payable to "THE ATOM USER GROUP" and sent to the address on the front page.

REAL TIME CLOCK FOR THE ATOM

The program listed below is an interrupt driven 24 hour clock for the Atom.

For those not familiar with interrupts a word of explanation might help. Essentially an interrupt is a means of making the microprocessor suspend operation of the piece of code it is currently executing, and to jump off to a predefined area of the computers memory to execute a piece of code called an interrupt routine. Once this code has been executed the micro then jumps back to its original place and continues as though nothing has happened. Of course, since micros run at very high speeds it can appear to the user that the micro is running two programs at the same time. Now it is possible to arrange the micro to be interrupted at regular intervals (say every second), and by further arranging for the interrupt routine to keep a track of how many it has received, and to output this count onto the VDU you can probably see that we now have the basis of a clock which will run and keep time no matter what the micro is doing (playing a game, etc.). In the case of the program below the time is continually displayed on the top right hand corner of the screen.

If you delve into the software you will find that the micro is interrupted 20 times per second by the VIA and counters for seconds, minutes and hours keep track of what the time is. Fortunately you can do anything you like on the Atom while the clock is running except SAVE or LOAD programs since the interrupt facility of the micro is disabled at this time. The actual device responsible for generating the interrupts is the 6522 VIA. This is a particularly powerful device and for those who are interested in knowing what else it can do they would be well advised to get a copy of the data sheet.

In its form below the routine lives in the space occupied by the floating point variables %@ to %R (it is about 10 bytes too long to live in the free space in block Ø RAM, and uses #AB to #AF for storage. The speed of the clock is determined by locations #B804 and #B805. The values in these locations can be changed if the CPU clock is not exactly 1 MHz.

Tony Dale (New Zealand)

LIGHT PEN FUNDAMENTALS

In Newsletter 6 we published a very simple light pen circuit diagram and gave a brief description of what would be required to configure it for the Atom. I have had some feedback from members which suggests that a little bit more detail might be helpful. Well here goes....

Essentially the picture on a television tube is made up of traces of light caused by a fast moving spot of light which is automatically scanned from the top left hand corner of the screen to the bottom right. The spot then 'flies' back to the top again and repeats the process many times in one second. It is only the eyes persistence of vision which allows us to view a complete picture. Now if it were possible to know just when the spot was about to set off on its journey and how long it will take to reach the bottom right hand corner of the screen then by using a light pen which produces a pulse when the spot passes it on the screen a little bit of calculation will tell us where the light pen is pointing on the screen. Fortunately the routine at FE66 (see page 6) tells us when the spot is returning to its starting position and this can be used to start a timer in the ATOM on the VIA (if you have one fitted). You then have to arrange for the signal from the light pen to interrupt the ATOM. Once this happens you stop the counter and from this you can work out where the pen is pointing.

REAL TIME CLOCK LISTING

```
 10 REM CLOCK
 20 DIM LL(10)
 30 H = %AB; M = H+1; S = H+2; C = H+3; T = H+4
 40 ?H=0; ?M=0; ?S=0; ?C=20
 50 U=%2800; REM INTERRUPT ROUTINE LIVES IN %@ TO %R.
 60 P.$21
 70 F.I = 1 TO 2; P=U
 80 [
 90:LL0 LDA %B804; DEC C; BNE LL7
100 TXA; PHA; TYA; PHA
110 LDA @20; STA C
120 SED
130 LDX @2; LDY @0
140:LL1 DEX; CLC
150 LDA M,X; ADC @1; STA M,X
160 CMP @%60; BNE LL2
170 STY M,X
180 TXA; BNE LL1; CLC
190 LDA H; ADC @1; STA H
200 CMP @ %24 CHANGE THIS TO 12 FOR 12 H. CLOCK; BNE LL2
210 STY H
220:LL2 CLD; LDY @9; LDX @6
230:LL3 DEX; DEY
240 STX T
250 TXA; LSR A; TAX
260 LDA H,X
270 BCC LL4
280 DEY; AND @%F
290 BCS LL5
300:LL4 LSR A; LSR A; LSR A; LSR A
310:LL5 CLC;ADC @%B0
320 LDX T; STA %8018,Y; BNE LL3
330:LL6 PLA; TAY; PLA; TAX
340:LL7 PLA; RTI
350 ]
360 N.I
370 P.$6
380 P.$12'THIS IS A 24 HOUR CLOCK.''
390 P.'YOU MUST INPUT THE TIME IN''
400 P.'HOURS, MINUTES AND SECONDS'''TO START IT.''
410 IN.'HOURS'A
420 ?H = (A/10)*16+A%10
430 IN.'MINUTES'A
440 ?M = (A/10)*16+A%10
450 IN.'SECONDS'A
460 ?S = (A/10)*16+A%10
470 REM SET UP IRQ VECTOR
480 ?%204 = U&%FF; ?%205 = (U/256)&%FF
485 REM COUNTER 1 VALUES.  CHANGE TO CHANGE SPEED OF CLOCK.
490 ?%B804 = 80; ?%B805 = 195
495 REM ENABLE INTERRUPT.
500 ?%B80B = %40; ?%B80E = %80+%40
510 END
```

THE ULTIMATE MEMORY TEST ROUTINE

This piece of software is a memory test routine to test each byte with every bit pattern from 00-FF HEX. First the byte at the address to be tested is saved and after the test is restored. If a failure is encountered 4 options are offered:

1. Increment bit pattern at the same address.
2. Increment test addres and restart bit pattern.
3. Restart test from the first address.
4. Abort.

Looking closely at the software you may think some of the branches are a little strange. This is because I have made the M/C code program completely portable. If the assembled code is *SAVED it can be *LOADed anywhere so the maximum amount of memory can be tested.

Zero page RAM is protected but the program location is not. If attempts are made to test the RAM that holds the program it will fail and may 'hang'. The headings show the address under test (ADDR), if a failure occurs the bit written (WR) and the bit read (RD) are displayed. The source program is about 1.5K long and the machine code assembly about 350 bytes.

Mr. Vials

```
 20    DIM VV15;F.N=0TO15;VVN=-1;N.
 30    F.N=0TO1
 40    DIM P-1
 50    P.$21
 60    [
 70    :VV0 JSR £FD69
 80    LDA @ 0
 90    LDX @ 12
100    :VV1 STA £79,X
110    DEX
120    BNE VV1
130    STA £E1
140    JSR £F7D1
150    ]
160    $P="RAM TEST";GOS.b;GOS.c
170    $P="ON ERROR INPUT:-";GOS.b;GOS.c;GOS.c
180    $P="0: REPT BIT PATN";GOS.b;GOS.c
190    $P="1: INC BIT PATN";GOS.b;GOS.c
200    $P="2: NEXT BYTE";GOS.b;GOS.c
210    $P="3: RESTART";GOS.b;GOS.c
220    $P="4: ABORT";GOS.b;GOS.c;GOS.c
230    $P="FROM PAGE(XX)";GOS.b;P=P-1
240    [
250    LDX @ £81
260    BNE VV3
270    :VV2 JSR £F7D1
280    ]
290    $P="TO PAGE(XX)";GOS.b;P=P-1
300    [
310    :VV3 LDA @ £3F
320    JSR £CD0F
330    LDY @ 0
340    JSR £F893
350    CMP @ 1
360    BNE VV3
370    LDA 0,X
380    CMP @ 4
390    BCS VV4
400    LDA @ 4
410    STA 0,X
420    :VV4 INX
430    INX
440    CPX @ £83
450    BEQ VV2
460    LDA £81
470    STA £8A
480    STA £82
490    CMP £83
500    BCC VV5
510    STA £83
520    :VV5 JSR £F7D1
530    ]
540    GOS.c;$P="ADDR WR RD";GOS.b;GOS.c
560    [
570    :VV8 LDY @ £FF
580    :VV6 INY
590    STY £89
600    LDX @ £89
610    JSR £F7F1
620    LDA (£80),Y
630    STA £ 84
640    LDA @ 0
650    STA £85
660    :VV9 LDA £85
670    :VV7 STA (£80),Y
680    LDA (£80),Y
690    CMP £85
700    BNE VV13
710    :VV10 INC £85
720    LDA £85
730    BNE VV7
740    STA £E0
750    :VV11 LDA £84
760    STA (£80),Y
770    CPY @ £FF
780    BNE VV6
790    LDA £81
800    CMP £83
810    BEQ VV15
820    INC £81
830    INC £8A
840    :VV12 LDA £81
850    BNE VV8
860    ]
880    [
890    :VV13 STY £87
900    STA £86
910    LDX @ £85
920    JSR £F7F6
930    DEX
940    JSR £F7F6
950    INC £E0
960    :VV14 JSR £FE94
970    JSR £F87E
980    CMP @ 5
990    BCS VV14
1000   CMP @ 4
1010   BEQ VV15
1020   TAY
1030   JSR £FFED
1040   LDX @ £89
1050   JSR £F7F1
1060   TYA
1070   STA £8B
1080   LDY £87
1090   CMP @ 0
1100   BEQ VV9
1110   CMP @ 1
1120   BEQ VV10
1130   LDA @ 0
1140   STA £E0
1150   LDA £8B
1160   CMP @ 2
1170   BEQ VV11
1180   LDA £82
1190   STA £8A
1200   STA £81
1210   BNE VV12
1220   ]
1230   REM**END & ABORT **
1240   [ :VV15 JSR £F7D1
1250   ]
1260   GOS.c;$P="END";GOS.b;GOS.c
1270   [ NOP
1280   LDA @ £80
1290   STA £E1
1300   RTS
1310   ]
1320   N.
1330   P.$6
1340   LINK VV0
1350   END
1360b  P=P+LEN P+1;R.
1370c  ?P=10;P=P+1;R.
```

NEWS.....NEWS.....NEWS.....NEWS.....NEWS.....

SOFTWARE FOR PRE-SCHOOL CHILDREN

Anyone interested in swapping software to amuse pre-school children contact Jim Percival, 26 Northfields Clowne, Chesterfield, S43 4BA. Alternatively 'phone 0246 811243.

ITEMS FOR SALE

Neil Harris (Tel: 0283 42558) has the following items for sale: A mixed bag of IC's including RAMS, DACS, PIO's, EPROMS £30.00 the lot. ATOM BBC Basic Board, boxed and unused £35.00. A card frame, guides, DIN connectors and sockets £30.00 the lot.

2K SCRATCH PAD RAM

CLARE COMPUTER COMPONENTS have brought out a board which fits inside the ATOM and fills the empty area of memory between #9800 and #9FFF. This is invaluable space that will not be corrupted by the ATOM Operating System. This area in particular is useful for those small machine code routines sometimes necessary to get round certain difficencies in the ATOM. The unit, say the manufacturers, is easy to fit, requiring no soldering and is supplied with full fitting instructions. Price is £12.00 + VAT (please check before ordering). The address to which all enquiries should be made is: 46 Bath Road, Stroud, Glous.

HOME COMPUTER CABINET

For those of you fed up with leads all over the place connecting your various pieces of equipment together, Macrol Cabinets have brought out a cabinet which will house the computer, tape deck, manuals etc. They can deliver anywhere in the country. Further information from 8 Bugle Street, Southampton, S01 0AJ.

ADD ON VIDEO BOARD

For those of you not satisfied with the resolution the Atoms graphic chip Intelgraph have brought out a board which effectively doubles the resolution. The board can produce 256*256 or 512*256 matrices requiring 8K and 16K of RAM respectively. It interfaces with the ATOM (or any other computer for that matter) via serial or parallel ports, at speeds of up to 19200 baud. The board is intelligent (6502) and executes instructions passed to it from the host computer. For example, there are single byte command to draw lines, arcs etc. from any point to any point. A single 5V supply is required. The unit produced s 1V composite video signal for a monitor or alternatively RF output for a domestic TV. The unit can be purchased completely built or as a bare board (all components are easily brought separately). Further information from 33 Bearsddn Cresent, Hinckley, Leicester, LE10 0SQ.

EPROM ERASER

At last a cheap EPROM ERASER at £19.95 for the Basic version or £24.95 for a version with a timer. The eraser can take up to 3 EPROMS or 1 CPU board with on-board EPROM. The unit is fairly small measuring 90*80*40 mm. Erasure times vary from 5 to 25 minutes depending on the device. Spare UV tubes are available. Further details from GROUND CONTROL, Alfreda Avenue, Hullbridge, Essex, SS5 6LT.

USEFUL SUBROUTINE ADDRESSES ON THE ATOM

Here is a list of some of the more useful routines to be found inside the ATOM ROM. I would be very interested to here of any others that readers have found. I will produce a list of these in a future Newsletter.

FD69 This produces a form feed and homes the cursor and clears the screen.

FE08 Scrolls the screen up one line.

FE22 Memory locations DE and DF point to the line where the next character will be printed. The routine at this location will actually erase the line pointed to by these memory locations.

FB8A This routine produces a delay of 1/10 second and is used by the ATOM to define the rate of character output if the repeat key is pressed.

FE71 Probably the most useful routine of the lot, it reads the keyboard and is handy in games programs. If no key has been pressed then#FF is returned in the Y register. If a key has been pressed the Y register contains the value ASCII-20H. Note that the routine does not recognise the CNTRL or SHIFT keys.

FE66 Another useful one. To get round the problem of noise on the screen when using graphics. This routine only returns when the flyback pulse from the TV scan is received, slows your program down only marginally.

F7FD Prints one space. There must be some use for this!

F802 This prints the contents of the Accumulator.

FD7D Homes the cursor.

F7D1 This routine prints a string of characters following the subroutine call.

C589 This converts a binary number to decimal and then prints it within the specified field width. The binary number to be converted should be in #16(LSB),#3 4,#43. The field width should be specified in #321.

FB83 This is primarily a delay routine and produces the delay by counting the number of flyback pulses specified in the X register before returning.

F893 This converts ASCII characters in the input buffer into a HEX address. The address is returned in the zero page location M (LSB) and M+1 (MSB). The routine exists on the first non ASCII character. Note that the X register points to the location M.

CDOF This is the start of the BASIC line editor and on calling will print the contents of A as a prompt and then wait for the user to type in a line of text which is then stored from #100 onwards. A carrage return terminates the line and returns control to the caller. An escape character at this stage returns control to BASIC.

USEFUL MEMORY LOCATIONS

As to be expected there are several locations in the ATOMS memory which have a particular significance, these are some of them:

DE & DF Point to the current cursor location (see above).
E7 This location defines what character is to be used for the cursor.
321-38C The BASIC variables @ to Z are held in these locations. Note that arrays are normally held at the TOP of your program.
12 Text space pointer.
08-0C Random number space.
23 & 24 Free space pointer.

EXPANDING THE ATOM

One of the major advantages of the Atom is that it can be readily expanded, "the personal computer that grows as you wish", to quote one of Acorn's leaflets. Whilst this is certainly true of internal expansion with more RAM, floating point ROM and VIA, I have found that off board or external expansion a great deal more tricky. However I have now overcome both hardware and software problems and I hope the hints below will help others avoid disappointments.

The External Bus

Acorn's literature describes how the bus comprises address lines (16), data lines (8) and various control lines together with 0 and +5V supplies. These are all brought together to a standard 64 way DIN connector (plug 6) together with the lines of both ports of the VIA (20). These lines are all paralleled to another connector (plug 7) to which one Eurocard can be fitted within the case. Both plugs are 'external' as far as the address map is concerned. I wanted to connect my Atom to the backplane of a cardframe (not Acorn's) so that I could easily add individual boards, for instance an A to D converter, but starting with an additional memory board.

This board was built from an Acorn kit but try as I could, I could not get the board to work, even though Acorn checked it for me and pronounced it OK. The problem was eventually traced to an error of Acorn's circuit diagram: the side A and B connections were shown reversed. My Atom board is issue 4 but the circuit diagram supplied with it was issue one so the problem has probably been corrected by now. Anyway, if you intend to use the external connecter, check the pin labelling very carefully by tracing the tracks on the board and comparing them with the circuit diagram. The row of pins nearest to the edge of the board is side A of standard DIN plugs and sockets. A similar error occurs on page 169 of the Atom manual.

The VIA lines are all on side A, not B. The Acorn bus is connected to side A on the Eurocard, so in effect it is necessary to swap A connections with the same numbered B somewhere in the interconnections.

My first interconnecting cable was about 1 metre long, with 59 tightly bunched wires and screened. Once the board was working in my system I found I had erratic operation of memory locations (a set of 200 nS chips were worse than 450 nS), shown as random failures of the RAM test program on page 92 of the Atom manual. This problem was cured by shortening the cable to ½ metre and removing the screening.

Finally, it is worth connecting together pins A1 and B1 and A32 and B32 on the Atom board, in the connecting cable, and on the external boards or backplane. If the PSU is in the cardframe, as mine is, these pins then provide a path for the 2.0 amps or so needed by the fully expanded Atom, in parallel with or instead of the rather unreliable and unsuitable power plug fitted to the standard Atom.

VDU Interface

A good deal of noise can be picked up by the UHF and IF circuits on a TV set from the unscreened bus cable, so it is important to keep the two as far apart as possible. The noise shows itself up as speckling and 'shotsilk' patterning of the display, and is especially irritating when half tones are being used. There is no such problem however when a monitor is being fed with composite Video direct from plug 4 of the Atom.

Brian Carroll

BOOK REVIEW - SPLITTING THE ATOM

The 8K of ROM used by the Atom's interpreter and operating system represent a lot of machine code and contains many useful routines, all of which are accessible by the user, enabling very efficient and compact programs to be written, taking full advantage of the Atom's capabilities. However, little has been written to tell you where they are and what they do. This is then the main aim of this book, to delve deep into the heart of the machine and uncover its secrets.

'Splitting the Atom' is described by the publishers as a manual for informed users, and that is indeed who the book is aimed at. Perhaps the type of person who wants to develop some sophisticated software of his own and needs to know the full facilities the machine offers. It won't actually teach you programming (except indirectly), but it will teach you how to use the Atom efficiently. To read this book is to embark upon an adventure - indeed there is something different on every page.

It actually comes as 80 pages of A4, spirally bound. There are 9 chapters and 4 appendices. After a brief introduction, it begins with an explanation of the various stacks used by the system, followed by an overview of the interpreters structure. Next comes details of the RAM used by the system and tables used for variable, array and lable stacks - you really need these since the way the Atom stores variables is complex. A description of every ROM routine then follows with entry points, register and stack usage, and suggestions for use. This is followed by a listing of both ROMS in assembler format. The next chapter gives 11 working with commented listings including a true CRC routine.

The final chapters cover a description of tape format, printer use and an outline of the memory mapped screens (with design charts for each mode). This section concludes with an exhaustive set of copy prevention routines. The four appendices provide a memory dump/edit program, a chart of the 6502 op-codes, mnemonics and addressing modes, a chart of ASCII and Control codes and a description of their own DISATOM ROM.

The book ends with a comprehensive index to the ROM routines. The manuscript was submitted to Acorn and carries Acorn's recommendation.

To conclude, armed with this book and 'Atomic Theory and Practice' you will know just about everything there is to know about the Atom.

Barry Pickles

Supplier: Focusplan Ltd.,
57 Westgate,
Cleckheaton,
West Yorkshire.

ATOM USER GROUP

Newsletter 10

Hello there once again! First of all many thanks to all of you who have contributed to this issue of the newsletter.If anybody out there has, or is, doing something interesting with their ATOM and think other users may be interested please do not hesitate in dropping me a line. A few notes are sufficient as long as they are accurate and complete.

I'm sure many of you who bought an ATOM some time ago are finding that it is very difficult to find anything useful to do with your investment. You probably have fulfilled your original aims which prompted you to purchase the machine in the first place wether these were simply to learn about micros or to use it as a game playing machine. It is very difficult on occasions to see what else you can do which will provide the same interest and excitement you probably experienced when you first unpacked it. In the last issue I expounded on the versatilty and power of the ATOM even though technically it is an 'old' micro. But dont forget that a TV set built in 1950 would be derided now, but it still performs the same function of it's modern day counterpart and it would provide no less information than its successors (it may be coloured and larger but that is just cosmetic).

So having I hope having convinced you that the technology shouldn't be the problem I hope I can appeal to sense of imagination to extract as much as possible from your investment. Given a basic micro with a moderate amount of RAM, graphics capability and back-up storage the potential exist in your hands to write software to perform countless applications. These can range from home finance, car and house maintainance records, shopping lists, gardening plans, sports statistics, weather records, cookery recipies, personal diaries, games, hobby records, scientific information (sophisticated programable calculator), If you have the ability to add some very simple hardware to your ATOM which allows it to interface to the outside world the possiblities become enormous. You could for instance read the positions of switches, measure temperature, pressure, light intensity, flow rates, sound levels, you could generate music or even speech again the list is only limited by your imagination After all the computers that run the Space Shuttle are based on exactly the same principals as your ATOM. They can accept input, proccess data and provide output. Your micro is the ultimate intellectual gymnasium. Are you going to sit there and become a mental fatty?

CONTENTS:

REVIEW OF THE ATOM DISC PACK - Geodfrey Manning

The addition of the disc pack to an ATOM computer system makes the machine look bigger from the programmers point of view but there are plenty of traps and downright bugs to trap the unwary. Also information on the DOS internal calls and on the structure of the catalogue is not widely known

SPECIFICATION

A good specification was used in the design of the operating system (DOS) including the virtual periperal facility which allows access to disc files at byte level, just as if they were arrays in memory. Unfortunately, a serious ommision is any sort of filing system that keeps an index of records within files, this facility is not supported even in a simple way.

It is therefore a shame that the DOS suffers from bugs that have crept in during its implementataion, plus errors and ommisions in its documentation (which of is poor quality). Acorn seem to have developed DOS from some previous system, pruning it down to fit the ATOM. It has to be said that if you are looking for a sophisticated disc system with index sequential facilities then this product may not be for you.

NEW ISSUE OF DOS

If you are buying the Disc Pack, be sure to obtain the DOS-2 issue ROM. This is characterised by a Cyclic Redundancy Check signature of HEX D89A (ATP page 93) This DOS corrects a major bug whereby file bounds were being ignored on sequential writes of consequtive bytes. The result of this was corruption of consequtive files on the disc.

DISC ORGANISATION

The floppy is divided into 400 sectors of 256 bytes each, with sectors 0 and 1 containing the disc catalogue. With 40 tracks on the disc, it is seen that there are 10 sectors per track and that the total formatted disc capacity is 100K inclusive of catalogue.

Although the disc format is single density, it is found that double density floppies are required in order to preserve data. Data integrity is enhanced by using a head cleaner frequently.

DRIVE UNIT

The pack contains an Olivetti FD501 drive. 48 tracks per inch, with a maximum unformatted capacity of 1/4 Mb , average access time of 0353s, maximum transfer speed 250kbits/sec. The drive itself is 6*15*20cm and weighs in at 1.35kg. The required +5V and +12V DC supplies are inside the disc pack box. The single head is on the underside of the floppy.

DOS COMMANDS: PROBLEMS AND THINGS TO WATCH OUT FOR

*COS: I wish it still existed but it doesn't. To switch from DOS to tape under software control requires you to keep a copy of the vector table from HEX 0200-021B. Make your copy and then call it *DOS. To return to COS, replace your copy of the vectors. You will need this facility when making disc to tape archives etc.

*TITLE: This must be used after formatting a fresh disc, otherwise the title comes out as random rubbish. This is a deficiency in FORM40.

*RUN: This will accept literal data ie parameters passed to the called program. For example running a program called MENU and telling it todays date is done by *RUN MENU 25/12/84 where the literal data is placed as a BASIC string at location HEX 0140. But what the manual does not tell you is that the string will get over-written by anything else in the progran which happens to use strings, so your first action must be to save it elsewhere. Example I=FIN MYFILE will put the string "MYFILE"
in location 0140. *RUN is actually not necessary. In the above example, *MENU 25/12/84 would do just the same thing, the rules about setting the array space and pointer TOP still apply exactly as under COS(see newsletter no. 6 page 2, preventing error 30).

*EXEC: It is little known that this will. only take data from the disc in the form of strings (ie ending in cariage return). You need to write a little utility using sequential SPUT to disc to get the strings on file in the first place.

*SPOOL is a facility to connect a disc file to the output stream, finished by the *SHUT command Anything printed out by the ATOM will be sent to the named file as well, using this command. More about this is to be found in a leaflet by ACORN ("Atom Disc Pack Spool,Exec and Verify).

Another serious bug. The final sector of the loaded file ought to be buffered. It isn't. Let me explain . Say a file contains just one byte (or any other size which is not a multiple of 256). The catalogue entry includes this file size. Lets us now load the file, perhaps to memory address HEX 8200. The single byte file goes in at HEX 8200 of course, but what about 8201 onwards?. This actually becomes corrupted. In fact, DOS has transferrred a whole sector of 256 byte to our memory regardless of file size. DOS should have first done the disc transfer to its internal buffer before copying out the file to the target memory.

This article will be continued in newsletter 11.

BEAR HARDWARE (BEARsoft) - ATOM micro support

Below are extracts from a long letter from Mike Barwise of BEAR HARDWARE, I'm sure some members will be encouragd by its contents.

BEAR HARDWARE was founded in 1983 by Mike Barwise to provide design and artwork for low volume PCB production and technical authorship. Support for the Atom was an early priority in this enterprise. It was evident that the Atom, the last of the generation of machines with "tweakable" hardware, was to be swamped by the squeaky and colourful, and that its inherent capability for expansion was going to remain unappreciated. The first development under the BEAR HARDWARE label was the BEARsoft EDITOR, a significant upgrade of the Acorn Wordpack, launched in March 1984. Previous to this an improved version of Acorn disk controller card was produced. This enables the Atom user to choose his own disk drive for the first time. Since then in association with Ken Dixon (software writer and radio amateur), we have been developing futher hardware, and looking for high quality software new and old.

The current position is as follows:

The EDITOR (upgraded Acorn wordpack) has been futher enhanced to improve efficiency with disc systems. The current version is V1.4. A review of V1.1 appears in the October '84 issue of ACORN USER.

We have commissioned a conversion software INTERFACE at a leading London typesetting house to allow DIRCET TYPESETTING using the EDITOR V1.4. The right to use the software is automatic to purchasers of the INTERFACE manual. The manual is in preparation, and enquiries are invited.

We are shortly taking over from SOFTWARE CLASSICS of Manchester their complete catalogue of Atom titles. We are in the final stages of negociations with Acornsoft for the right to supply their original Atom titles. We have intended to develope an upgraded Atom operating system with a 2-3K permanent extension and rationalised ROM space handling to allow efficient use of RAM/DISK as an alternative to the rare 2532.

In view of the obsolescence of the 2532 EPROM in the very near future, we are developing a new disk controller compatible with the Atom, and other hardware upgrades to support disk. Additional hardware projects include: tidying up the Atom memory map, GOTO page 7

INTERFACING STEPPING MOTORS TO THE ATOM

This article is intended to provide some tips on interfacing stepping motors to the Atom. Fist of all what are stepping motors and their uses? Most people are familiar with motors which rotate smoothly in one direction at some predefined speed. These are fine as far as they go but are not suitable for applications where we wish to move something accurately to a predefined position or where we wish to move forwards or backwards at a known speed. Stepper motors overcome these problems and have been designed specifically with digital control in mind. A typical stepper motor is quite small (about 2 inches in dia.) has one output shaft, 4 control wires and 1 earth return wire. Each control wire is responsible for energising a particular phase of the motor. By energising the 4 phases in the correct sequence it is possible to make the output shaft rotate. Note that the rotation is not smooth but consists of small steps each one corrsponding to a change in state of the phases. The rate at which the state of the phases are changes defines the rate of rotation of the motor. The stepping resolution varies but typically would be about 1 degree per step. The maximum speed of the motors is about 60 rpm. In this context it is important to realise that the output torque of stepper motors drops dramatically at the top end of their speed range.

To interface this type of motor to the Atom requires the VIA to be installed. This as two 8 bit wide ports which can be used for controlling two stepper motors. We also require some simple circuitry to increase to current driving capability of the VIA's outputs so thay can drive the phases of the stepper. Of course we need a stepper motor and finally some software to drive the whole thing. Initially it is an interesting project simply to get the motors running from the Atom without having any particular application in mind, dreams about graph plotters and robots are best left until this stage has been achieved.

You may be wondering where on earth you can buy a stepper motor. Commercial ones are very expensive but alternative suppliers are available who deal in surplus MOD stock. These are available from the firm mentioned at the end of the article for about 12 pounds which is very cheap. The specification of these motors is as follows: max phase current 300mA, torque 20oz/in, no. of steps per rev 200, phase voltage 5-30V. As stated above we can control the motor by setting up the outputs of one of theports on the VIA chip. The sequence in which these outputs must change is as follows:

	PHASE			
	1	2	3	4
a	1	1	0	0
b	0	1	1	0
c	0	0	1	1
d	1	0	0	1

Note that it is possible for the motor to be rotated backwards by simply send the phase information out in the reverse sequence.

GOTO page 10

REVIEW ROSS SOFTWARE UTILITY ROM

In order to use this ROM all you need is an ATOM with a spare utility ROM socket. In brief the facilities offered are , a flashing cursor, 33 new commands and the provision of a 1200 baud cassette interface. After initialisation the chip is booted up by typing "LINK 44992". From this point all the facilities of the chip are available, these are described in detail below.

*DATA,*READ — Very useful indeed. These give a full implementation of Microsofts Basic commands quotes are required round string data.

RESTORE N — This tells the Atom to read the next data element from the beginning of the data list, starting at line N. This addition of line numbers to this command allows easy access to several data lists.

RENUMBER X,Y — Renumbers the lines of a program, starting with line X stepping Y between lines. It also renumbers all GOTO's, GOSUB's and RESTORES.

AUTO X,Y — Provides automatic line numbering when entering programs, the definition of X and Y is as above. If the file already exists a warning beep is given.

BSAVE name — Saves a BASIC program so that it can be auto run with *RUN. Note that it does not work with programs that contain DIM statements.

VAR — This the current values of all integer variables A-Z.

APPEND name — Merges a program on cassette with one on the computer.

VERIFY name — Compares a program on tape with one in memory. As each block is verified correctly a display similar to *CAT is produced. An error is given if there is a mismatch.

POINT X,Y,P — X,Y denotes the position of a graphics pixel. If the point is set (ie white) then P=1 otherwise P will be zero (P can be any variable

BLOCK N,X,Y,X1,Y1This command draws a block as follows. N is 1,2 or 3 to set,clear or invert the block respectivly. X,Y is the position of the bottom left hand corner. X1 is the horizontal length of the block and Y1 the vertical length of the block.

SHAPE,*TABLE The command *SHAPE N,X,Y draws a complex user definable shape (number N), at X,Y the direction up,down,left,right are stored in a table whose base address is set by the command TABLE.

ZERO This command zeros the integer variables.

*BLEEP,*BPOFF When the chip is initialised all cassette operations end by bleeping endlessly until a key is pressed (so you do'nt have to sit staring at the screen waiting for the return of the prompt). *BPOFF switches off the facility, *BLEEP turns it back on.

TONE Generates a tone of pitch N (1-256) and duration D (1-65536).

TAPE s When the chip is initialised the cassette operating system works at 1200 baud. However if old or bought programs (saved at 300 baud) need to be loaded the speed can be reset by typing *TAPE1 (300 baud) *TAPE2 sets the baud rate to 600 and *TAPE3 to 1200 baud.

STOP A debugging tool. This causes the message STOP (line number), and waits for a key to be pressed before continuing.

POP Allows exit from the current subroutine without a return so that a "too many GOSUBs" error is not produced if the subroutine is endlessly called, and jumped out of.

DELETE X,Y Erases lines X to Y inclusive.

PACK Removes all insignificant spaces from programs.

KEY N This is an INKEY command. If no key then N = null.

SCREEN X X is a number between 0 and 511. The cursor is placed at this position on the screen.

FIND In my opinion the most useful command, it list all the lines containing the given string.

MC Another useful command allowing entry into the machine code monitor. This has three commands: memory change,hex dump and text dump.

STRG Another useful command, printing "string" at an X,Y position on the graphics screen. This command works in any mode (0-4).

BSTRG As above but prints out the string stored at 0140 in memory.

CHAR C,X,Y Prints the character with ASCII "C" at X,Y in all graphics modes.

*KBEP Generates repetitive bleeping from the loud speaker until any key is pressed.

For machine code programmers, at #A000, there is an index to the chip, consisting of a list of each instruction (minus the "*") and its start address in the chip. Looking through this (using *MC of course) shows that there is an unused instruction *VDU. Typing this instruction will cause a jump to 2800 and so can be used to execute the users own routines.

The price for these facilities is approx 15 pounds (check before ordering). The mail order address is:

ROSS SOFTWARE, 44PREMIERE AVENUE, GRAYS ESSEX RM16 2SD

Bear Hardware continued........
releasing 5.5k of currently inaccesible memory, and a conversion card to put RAM in #D000 and #A000 utilities spaces. We are commissioning new titles for the Atom (maths,comms,scientific and good text based adventures). If you have any goodies drop us a line. We promise not to pirate your ideas, and if we can market the software we pay 20% per sale. To keep the Atom alive and well, the most important thing is contact. We can only help you if you tell us your out there. Our address for all enquiries is BEAR HARDWARE 68,Harmondsworth Lane, Harmondsworth, Middlesex, UB7 0AA

REVIEW OF HARDWARE OPTIONS AVAILABLE FROM CLARE COMPUTERS

INTRODUCTION

The Clare Computer Component System was specifically designed for the Atom. It is an ideal system for program development and for permanent or semi permenant storage of programs and data. The modules are paged (16k/page) in the extension RAM and are simply switched using port B of the VIA chip. Although the modules were intended as a stackable system all of the separate units can be used alone so the system can be built up to suit personal requirements.

8K CMOS RAM+8K ROM - MASTER

This module has beeen designed as a program developement tool and for regulary used programs. The 8k CMOS RAM can be supported by a battery so that it will retain its data without a mains supply, this of course means that you don't have to bother saving partially written programs. There are also two sockets for 2532 EPROMS so that your favourite programs are available on power up. Ideal for household accounts or a popular game.

This module also contains the logic for paging seven other modules and has power-in sockets for battery back-up and a secondary supply for the system if required A single module will run direct from the Atom. The module can be plugged straight into the Atom external extension socket if used alone or can be connected via a ribbon cable to form the base for stacking the system.

8K CMOS RAM+8KROM - SECONDARY

This module is as above but it is modified to fit the system mid-stack and it has no paging logic.

16K EPROM MODULE

This module is for futher storage of much used programs and has sockets for two 2532 (4K) and four 2716 (2k) EPROMS. The sockets are not the zero force insertion type but EPROMS can easily be changed with care.

Besides the above products Clare also produce an EPROM programmer and a battery back-up module. The battery back up unit is very small and takes its power from the master unit and trickle charges the battery which should apparently last 4 years per 8K.

Prices for these products are as follows, 8k master and 8k secondary 50.00 each the EPROM module is 27.50, the EPROM programmer is 52.50 and the battery back up unit is 6.00. Please check these prices however if you wish to order any of the above. VAT must be added to all orders. The address for information on any of the above is: Clare Computer Components, Freepost GR 1271,Stroud,Glos. GL5 3JL

USEFUL ROUTINE USING THE ACORN WORD-PROCESSOR

I have developed the following routine which requires the word processor to be fitted at #A000. I'm sure other users will find it very useful. It consists os a short subroutine which allows printing of the full word processor characters in any of the graphics modes, it is written in Basic as I am not clever at writting machine code which would certainly improve it substantially. Use the program like this

1. Put the characters wanted in $A
2. Place the location you want to start printing at in variable "L". The top left of the screen in all modes is #8000 and there are up to 192 lines to start printing at, a character is 8 lines high (about 0.75 inches in mode 1).
3. Store the horizontal spacing in variable H.
4. Store the vertical spacing in the variable V.
5. When you want to print $A simply execute a GOS.a statement.

How it works. The word processor stores its character pattern from #AD00 to # AFFF in groups of eight bytes per character. This subroutine merely locates the first character in $A and counts a multiple of eight from #AC00 it then pokes that byte to the start address stored in "L". It then pokes the next seven bytes below "L", it repeats the routine until the end of $A-1 this prevents trying to print the end of string marker "13". The spacing both vertical and horizontal may be positive or negative allowing some interesting labelling of diagrams. The reults of printing in mode 1 are very useful for anyone with poor eyesight. I am sure that other usres will be able to make many alterations to the routine. One such improvement would be to use an equation to calculate the position of the next print position to allow printing along curves etc.

The basic routine:

```
                    1000aF.X=0TOLENA-1;Z=A?X;Z=(#AD00+8*(Z-
32));D=0;F.Y=0TO7     1010?D=Z?Y;D=D+32;N.Y;L=L+H+(V*32)
1020IF L%32=0;L=L+(SGNV*256) 1030N.;R.
```

Below is a short routine using my program.

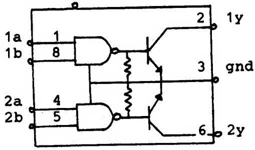
```
 10 DIMA64,B64,C64,E64
 20 $B="  LONGEST  ";$C="ADJACENT";$E="OPPOSITE"
 30 CLEAR 4
 32          $A="Triangles          and          their
sides";L=#80A2;H=1;V=0;GOS.a
 40 MOVE10,10;PLOT2,170,0;PLOT2,0,170;PLOT5,10,10
 50 $A=$B;H=2;V=-16;L=#9421;GOS.a
 70 $A=$e;H=0;V=16;L=#8458;GOS.a
 80 $A="@o";L=#9525;H=1;V=-8;GOS.a
 81 F.Q=0TO9;L=#9421;H=2;V=- 16;$A=$B;GOS.a
 82 L=#9421;H=2;V=- 16;$A="HYPOTENUSE";;GOS.a;
 83 F.X=0TO1000;N.;N
 85 $A="NOT BAD ??";L=#9009;H=1;V=0;GOS.a
 90 E.
 1000aF.X=0TO LENA-1;Z=A?X;Z=(#AD00+8*(Z- 32));D=0;F.Y=0TO7
 1010 L?D=Z?Y;D=D+32;N.Y;L=L+H+(V*32)
 1020 IF L%32=0;L=L+(SGNV*256)
 1030 N.;R.
```

A few points worth mentioning. To stop "gash" appearing on the screen by typing "P.$21" this eliminates any prompts. Inputs can of course be used ie "IN.$A" To get the VDU on again simply type CNTRL F, its quicker than "P.$6". Paul Alcock SOMERSET

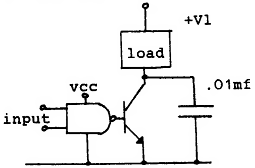
Interfacing stepper motors to the Atom cont.....

It is the responsibility of the software to provide the inter step delays. Flexibility is possible here in that by ramping down the delay between steps we can effectivly create acceleration, deceleration of course being the reverse effect. Since we know precisely the angle through which the motor has stepped we know its position by simply keeping a count of the pulses we have applied.

Below is the device I used to drive this particular stepper from the VIA output ports on the ATOM. You may find difficulty in obtaining this part but the circuit is very simple to construct using different devices. The basic principal in the design is "TTL in, lots of milliamps out"



LAMBDA LPD4101 Driver Chip



Basic circuit

Steppers available from STUARTS of READING 110 Wykeham Road, Reading.